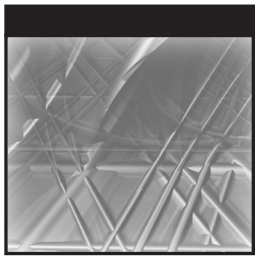
Troubleshooting Campus Networks

Practical Analysis of Cisco and LAN Protocols

Troubleshooting Campus Networks

Practical Analysis of Cisco and LAN Protocols



Troubleshooting Campus Networks

Practical Analysis of Cisco and LAN Protocols

Priscilla Oppenheimer
Joseph Bardwell

Wiley Publishing, Inc.

Publisher: Robert Ipsen
Editor: Carol Long
Developmental Editor: Adaobi Obi
Managing Editor: Micheline Frederick
Text Design & Composition: Wiley Composition Services

Designations used by companies to distinguish their products are often claimed as trademarks. In all instances where Wiley Publishing, Inc., is aware of a claim, the product names appear in initial capital or ALL CAPITAL LETTERS. Readers, however, should contact the appropriate companies for more complete information regarding trademarks and registration.

This book is printed on acid-free paper. ♾

Copyright © 2002 by Priscilla Oppenheimer and Joseph Bardwell. All rights reserved.

Published by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4470. Requests to the Publisher for permission should be addressed to the Legal Department, Wiley Publishing, Inc., 10475 Crosspointe Blvd., Indianapolis, IN 46256, (317) 572-3447, fax (317) 572-4447, E-mail: permcoordinator@wiley.com.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Library of Congress Cataloging-in-Publication Data:

ISBN: 0471-21013-7

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Contents

Acknowledgments **xvii**

Chapter 1 Introduction **1**
Why We Wrote This Book 1
Guaranteed Not to Rust, Bust, or Collect Dust 2
Audience and Scope 3
Cisco Certifications 3
The NAX Certification Program 4
Organization 5
Our Web Site 6

Chapter 2 Troubleshooting Methods **7**
Challenges in Today's Networking Environment 7
Using a Systematic Troubleshooting Method 9
Using the OSI Model for Troubleshooting 10
Generic Problem-Solving Models 12
Fundamental Network Troubleshooting Considerations 13
Cisco's Troubleshooting Method 14
Define the Problem 15
Gather Facts 15
Consider Possibilities 16
Create an Action Plan 16
Implement the Action Plan and Observe the Results 16
Document the Results 17
Proactive Troubleshooting and Baselining 17
Documenting Your Network 18
Documenting Network Names and Addresses 19
Tools for Network Documentation 20
Documenting Switched Networks 21
The Cisco Discovery Protocol 22

Protocol Analysis 23
WildPackets Protocol Analyzers 24
Using a Protocol Analyzer in a Switched Network 24
Understanding Network Traffic 27
Reliable Versus Unreliable Protocols 27
Terminal/Host Traffic 28
Peer-to-Peer Traffic 29
Client/Server Traffic 30
Server-to-Server Traffic 32
Distributed Computing Traffic 32
Statistical Monitoring 32
Simple Network Management Protocol and Remote Monitoring 34
WildPackets Statistical Tools 35
Simulation and Modeling 35
Active and Reactive Troubleshooting 36
Tools for Troubleshooting the Physical Layer 36
Tools for Troubleshooting Above the Physical Layer 37
Ping 37
Trace-Route 41
Other Troubleshooting Commands 45
Cisco IOS Show Commands 45
Cisco IOS Debug Commands 48
Summary 50

Chapter 3 Troubleshooting and Analyzing Ethernet Networks 53
Ethernet History and Architecture 53
Ethernet Topologies 54
Ethernet Physical Layer 56
Protocol Analyzers and Ethernet Media 59
Signal Encoding 59
Manchester Encoding 59
MLT-3 Encoding 60
Other Signal Encoding Methods on Ethernet Networks 62
Clock Synchronization 62
Ethernet MAC Layer 63
Carrier Sense Multiple Access with Collision Detection 63
Collision Domains 65
Bit Length and Measurement of a Collision Domain 65
Collision Domains in 100-Mbps Ethernet 66
Collisions on Networks with Hubs and Switches 67
Full-Duplex Operations 68
Full-Duplex Protocol Analysis 69
Autonegotiation 70
Flow Control on Full-Duplex Links 72
Enhancements to Ethernet Media Access Control for Gigabit Ethernet 73

Ethernet Frames 75
Fields in an Ethernet II or IEEE 802.3 Frame Header 76
Destination Address 76
Source Address 78
Length or EtherType 78
Logical Link Control Frame Formats 80
Ethernet or IEEE 802.3 Frame Footer 83
Ethernet Frame Sizes 84
Configuring and Troubleshooting Ethernet Frame Sizes 85
Maximum Transmission Unit 87
Tunneling and Frame Lengths 87
VLAN Tagging and Baby Giants 88
Collecting and Understanding Ethernet Performance Data 88
Remote Monitoring 88
The Cisco Show Interface Ethernet Command 89
Bandwidth 90
Delay 91
Utilization 92
Reliability 93
Collision Rate 94
Frame Corruption 95
Broadcast and Multicast Rates 96
Cisco's Broadcast Suppression Feature 97
Summary 98

Chapter 4 Troubleshooting and Analyzing IEEE 802.11 Wireless Networks 99
Chapter Scope and Depth 99
Things That Are Beyond the Scope of This Chapter 100
A Primer on Wireless Networking 102
Wireless Networking Fundamentals 102
What Is a Wireless Network? 103
The Challenge of Wireless Network Design 103
Protocol Analysis in Wireless Networks 105
Wireless Networking Technologies 105
Narrowband Radio Transmission (Radio LAN) 106
Frequency Hopping Spread Spectrum (FHSS) 106
Direct Sequence Spread Spectrum (DSSS) 106
Infrared (IR) 107
Bluetooth and the Personal Area Network (PAN) 107
Conflicts between Bluetooth and 802.11 Networks 108
Other Wireless Standards 108
Organization of the WLAN Environment 109
The Basic Service Set (BSS) 109
The Extended Service Set (ESS) 110
The Timing Synchronization Function (TSF) 112

Station Initialization Behavior 112
Channel Selection 113
Authentication 115
Association to the BSS 115
Analysis of the Initialization Process 115
Packet Acknowledgment 116
802.11 Media Access Control 117
Interframe Spacing 117
The Network Allocation Vector (NAV) 118
The Request to Send/Clear to Send Mechanism 119
Synopsis of the 802.11 Environment 119
Radio Frequency Transmission 120
Frequency Allocation 121
Mathematics for RF Engineering 122
The Decibel Unit of Measurement 123
A Synopsis without Any Complicated Math 123
Understanding Decibels 124
Important Things to Remember about Decibels 126
Specifications Involving Gain or Loss 126
The dB Milliwatt 127
Relationships between Metrics 127
The Wonder of Logarithmic Calculations 128
Applying the Principle of Logarithmic Subtraction 129
Concluding Thoughts on Logarithmic Subtraction 130
Application of dB and dBm Measurements 130
Free Space Propagation 130
Multiplying Logarithms Is Like Exponentiation 131
Free Space Path Loss Conclusions 133
Environmental Factors That Affect 802.11 Transmission 133
Reflection 134
Absorption 134
Refraction 134
Diffraction 134
Concepts for Site Survey Troubleshooting 134
Clear Channel Power Assessment 135
Limitations on RF Signal Transmission 136
Shannon's Channel Capacity Theorem 136
Gaussian Noise 137
Multipath Transmission 137
Real-World Path Loss 138
Calculating Real-World Path Loss 138
Antenna Positioning for Maximum Coverage 139
Protocol Analyzer Reporting of Signal Strength 140
Determining the Significance of Vendor Specifications 141
Output Power 142
Receiver Sensitivity 143

The Impact of Environmental Noise 143
Network Utilization 145
Signal Strength, Bad Packets, and Network Utilization 145
Checksum Errors 146
Site Survey Techniques for WLAN Troubleshooting 146
Estimating Effective Range 146
AP Placement Considerations 147
Troubleshooting Network Design Problems 148
Large Packets and Collisions 149
Numerous Simultaneous Users and Collisions 149
Practical Limits on User Community Size 149
Configuration Settings 151
IP Address 151
ESSID 151
Fragmentation Threshold 151
Request to Send (RTS) Threshold 151
Authentication Type 152
Wired Equivalent Privacy (WEP) Key and Passphrase 152
Understanding the 802.11 Packet Decode 152
Summary 154

Chapter 5 Troubleshooting and Analyzing the Spanning Tree Protocol 157
Poetic Interoperations 157
Transparent Bridging 158
Bridging Tasks 162
Bridging Loops 162
STP Behavior 165
Bridge Protocol Data Units 166
Protocol Analysis of BPDUs 166
STP Convergence 172
Electing the Root Bridge 172
Electing Root Ports 174
Electing Designated Ports 174
Selecting Bridge Ports for the Spanning Tree 175
Port States 176
The Topology Change Process 177
Proactive Troubleshooting of STP 180
Documenting Your Switched Network 182
Monitoring STP 182
Logging STP Events on Cisco Switches 184
Logging STP Events on Cisco Routers 185
Reactive Troubleshooting of STP 186
One-Way Connectivity 188
Reducing Startup Delay on Cisco Switch Ports 188
Cisco's Portfast Feature 189

Optimizing Spanning Tree Implementations 192
Uplink Fast and Backbone Fast 192
Load Sharing 193
Selecting the Root Bridge Deterministically 194
Configuring Bridge Priority 197
Summary 197

Chapter 6 Troubleshooting and Analyzing Virtual LANs 199
VLAN Frameworks 199
VLAN Definitions 200
VLANs and Non-VLANs 200
The Design of a VLAN 201
Interconnected Switches 202
Protocol Analysis in a VLAN Environment 202
VLAN Memberships 204
Configuring VLANs 204
Assigning a VTP Domain 205
Creating a VLAN 206
Assigning Ports to VLANs 207
Verifying VLAN Configuration 208
VLAN Trunks 210
ISL and 802.1Q Comparison 212
Routers and VLAN Trunks 213
Cisco's Inter-Switch Link 217
Dynamic Inter-Switch Link Protocol 218
DISL Frames 220
IEEE 802.1Q 222
Configuring 802.1Q 224
Per-VLAN Spanning Trees 224
Troubleshooting Cisco's VTP 227
VTP Names and Passwords 228
VTP Pruning 228
Analyzing and Monitoring VTP 229
Summary 235

Chapter 7 Troubleshooting and Analyzing Campus IP Networks 237
TCP/IP History 237
The TCP/IP Protocol Stack 238
The Internet Protocol 241
IP Protocol Analysis 241
The IP Type of Service or DS Field 243
IP Fragmentation and Reassembly 246
IP Time to Live 252
The IP Protocol Field 254
IP Options 254
IP Addressing 260
Moving the Prefix Boundary to the Right (Subnetting) 261
Claude Shannon and Boolean Logic 264

Moving the Prefix Boundary to the Left (Supernetting) 265
Address Resolution Protocol (ARP) 268
Proxy ARP 269
Reverse Address Resolution Protocol (RARP) 270
Dynamic Host Configuration Protocol 270
DHCP and Routers 273
Monitoring and Troubleshooting DHCP 276
Private IP Addresses 277
Network Address Translation (NAT) 278
IP Multicast Addresses 279
The Internet Group Management Protocol (IGMP) 280
Multicast Routing Protocols 281
IPv6 282
IPv6 Protocol Analysis 284
IPv6 Extensions 285
IPv6 Autoconfiguration 286
Summary 287

Chapter 8 Troubleshooting and Analyzing Campus IP Routing Protocols 289
Host Routing 289
The Hot Standby Router Protocol 292
Monitoring and Troubleshooting Host Routing 292
Static and Default Routing 293
Monitoring and Troubleshooting Static and Default Routes 294
Dynamic Routing 295
Distance-Vector Versus Link-State Routing 298
Distance-Vector Routing Protocols 298
Link-State Routing Protocols 300
Using Multiple Routing Protocols 301
Integrated Routing and Bridging 302
General Comments on Troubleshooting IP Routing 303
Sample Network Used in Protocol Analysis Examples 304
Routing Information Protocol (RIP) 305
RIP Protocol Analysis 306
RIP Timers 309
RIPv2 310
Cisco Show and Debug Commands for RIP 311
Interior Gateway Routing Protocol (IGRP) 312
IGRP Protocol Analysis 313
IGRP Triggered Updates and Poison Reverse 315
Cisco Show and Debug Commands for IGRP 318
Enhanced IGRP 319
EIGRP Protocol Analysis 320
EIGRP Queries 325
Cisco Show and Debug Commands for EIGRP 327

Open Shortest Path First (OSPF) 329
OSPF Network Architectures 330
OSPF Protocol Analysis 331
Building an Adjacency 333
Cisco Show and Debug Commands for OSPF 338
Border Gateway Protocol (BGP) 340
BGP Protocol Analysis 340
Cisco Show and Debug Commands for BGP 343
Summary 344

Chapter 9 Troubleshooting and Analyzing TCP, UDP, and Upper-Layer IP Protocols 345
Upper-Layer IP Protocol Analysis 346
Transmission Control Protocol 346
TCP Protocol Analysis 347
TCP Port Numbers 348
TCP Connection Establishment 350
TCP Reliable Delivery and Flow Control 354
TCP Delayed Acknowledgments 358
TCP Slow Start 359
TCP Window Size Analysis 359
Identifying TCP Performance Problems 362
TCP Connection Termination 364
TCP Checksum 365
User Datagram Protocol 366
UDP Protocol Analysis 367
UDP Port Numbers 368
Application-Layer Protocols 368
Domain Name System 369
DNS Protocol Analysis 371
Service Location Protocol 375
Hypertext Transfer Protocol 377
HTTP Proxy Servers 378
HTTP Protocol Analysis 379
File Transfer Protocol 381
FTP File Types and Structures 382
FTP Protocol Analysis 383
FTP Active Versus Passive Mode 387
Protocols for Electronic Mail 392
SMTP Protocol Analysis 393
POP3 Protocol Analysis 395
Summary 396

Chapter 10 Troubleshooting and Analyzing Campus IPX Networks 399
Novell NetWare Concepts 400
IPX Addressing 401
IPX Packets 403
Ethernet Frames in IPX Environments 406

NetWare Core Protocol 409
Watchdog Spoofing 412
IPX Service Advertising Protocol 414
SAP Filters 417
Get Nearest Server 419
Cisco Routers and Get Nearest Server 420
IPX Routing 421
IPX Routing Information Protocol 422
Enhanced Interior Gateway Routing Protocol for IPX 424
NetWare Link Services Protocol 427
IPX Networks in Transition 430
Troubleshooting IPX 433
Applying Your Protocol Analyzer 434
IPX Ping and Trace 434
Show IPX Route 435
Show IPX Servers 436
Show IPX Interface 436
Show IPX Traffic 437
Show IPX EIGRP 438
Show IPX NLSP 439
Debug IPX Packet 439
Debug IPX Routing 439
Debug IPX SAP 439
Summary 440

Chapter 11 Troubleshooting and Analyzing Campus AppleTalk Networks 441

AppleTalk Concepts 441
AppleTalk Architectures 442
AppleTalk Layering 443
AppleTalk Addressing 446
AppleTalk Addresses on a Cisco Router 447
AppleTalk Address Resolution Protocol (AARP) 448
Dynamic Addressing 448
Datagram Delivery Protocol (DDP) 452
Name Binding Protocol (NBP) 454
Apple Filing Protocol (AFP) 455
AFP over TCP 457
AppleTalk Routing 459
Routing Table Maintenance Protocol (RTMP) 460
End-Node Routing 463
Zone Information Protocol 464
AppleTalk Update-Based Routing Protocol 465
EIGRP for AppleTalk 468
Controlling AppleTalk Traffic 473
Static Routes 473

AppleTalk Access Lists 474
Filtering Cable Ranges 474
Filtering Zones 475
Macintosh Networks in Transition 475
Mac OS X 476
WebDAV 478
Service Location Protocol (SLP) 478
Mac OS X Initialization Traffic 478
Mac OS X Server Network Management Tools 480
Troubleshooting AppleTalk 480
AppleTalk Ping (Echo) 482
Cisco AppleTalk NBP Testing 483
Cisco IOS Show Commands for AppleTalk 484
Cisco IOS Debug Commands for AppleTalk 486
Summary 487

Chapter 12 Troubleshooting and Analyzing Windows Networking 489
Windows Networking Concepts 489
The NetBIOS Basis for Windows Networking 491
The History of NetBIOS 491
NetBIOS Function Calls 491
The Importance of NetBIOS 492
NetBIOS Consistency across Different Implementations 494
NetBIOS Naming Conventions 494
Qualifier Byte = 0x00 495
Qualifier Byte = 0x03 495
Qualifier Byte = 0x05 495
Qualifier Byte = 0x06 496
Qualifier Byte = 0x1B 496
Qualifier Byte = 0x1C 496
Qualifier Byte = 0x1D 496
Qualifier Byte = 0x1E 496
Qualifier Byte = 0x1F 497
Qualifier Byte = 0x20 497
Qualifier Byte = 0x21 497
The 0x0102_MSBROWSE_0x0201 Name 497
The Significance of NetBIOS Naming 498
NetBIOS Implementation Differences 499
A Windows Internet Name Service Query Carried on UDP 500
A TCP NetBIOS Session Setup Request 501
TCP NetBIOS Data 502
NetBEUI Data Exchange (A Browse Packet) 504
IPX Name Query 506
IPX NetBIOS Data 507
Concluding Thoughts on NetBIOS Terminology 508

NetBIOS Name Management Mechanisms 508
Name Registration 508
Name Resolution 509
Name Refresh 510
Name Release 510
Management of NetBIOS Names with NetBIOS/TCP 510
Troubleshooting NetBIOS Naming Problems 512
NetBIOS Data Movement 513
Session Behavior with NetBIOS/TCP 514
Reliable Data Transport with NetBIOS 515
The Workgroup Model and the Domain Model 518
Analysis of a Workgroup Authentication Operation 518
Authentication in the Windows NT Domain Model 521
NT Registry Parameters Controlling Replication 522
The Browse Protocol 522
Maintaining the Browse List 523
Browsing with Multiple Protocols 524
The Windows 2000 Model 525
Summary 527

Chapter 13 WAN Troubleshooting for LAN Engineers 529
WAN Concepts 529
WAN Standards 530
Understanding WANs from a LAN Management Perspective 532
WAN Components 532
End-to-End WAN Considerations 533
Troubleshooting WANs from a LAN Perspective 534
Troubleshooting WANs from a Router Interface Perspective 536
Provisioning WAN Capacity 537
WAN Technologies 539
Leased Lines 539
High-Level Data Link Control (HDLC) 539
Troubleshooting Cisco's HDLC Implementation 540
Point-to-Point Protocol (PPP) 541
Troubleshooting PPP 543
Frame Relay 544
Frame Relay Virtual Circuits 544
Frame Relay Congestion Control 545
Frame Relay Inverse ARP 546
Frame Relay Local Management Interface 547
Troubleshooting Frame Relay 547
Integrated Services Digital Network 551
ISDN Layer 1 552
ISDN Layer 2 553
ISDN Layer 3 555
Troubleshooting ISDN BRI Problems 556

Asynchronous Transfer Mode (ATM) 557
ATM Virtual Circuits and Adaptation Layers 558
Mapping a Protocol Address to an ATM PVC 559
ATM Quality of Service Specifications 560
Troubleshooting ATM 561
Chapter and Book Summary 562

References **565**
Standards Documents 565
Web Sites and White Papers 567
Books 568
Books on LANs 568
Books on WANs 568
Books on TCP/IP 568
Books on Network Technology, Design, and Troubleshooting 569

Index **571**

Acknowledgments

We would like to thank the professionals at Wiley who helped make this book a reality, especially Carol A. Long, Executive Acquisitions Editor, Adaobi Obi Tulton, Assistant Developmental Editor, and Micheline Frederick, Associate Managing Editor. In addition, each of us has many other people to acknowledge.

Priscilla Oppenheimer: I would like to thank the many protocol experts with whom I have worked over the years. Thanks to Howard Berkowitz, a helpful and wise computer networking wizard, and Marty Adkins, a CCIE who is always ready with a comprehensible answer. Thanks also go to my colleagues on the Group Study mailing list who have answered numerous technical questions, including Paul Borghese (the moderator), Chuck Larrieu, Leigh Anne Chisholm, David Madland (MADMAN), Pamela Forsyth, Jenny McLeod, John Neiberger, Rita Puzmanova, Paul Werner, Karen Young, and many others. Special thanks to Kevin Cullimore and Brant Stevens for their Novell NetWare expertise.

I would like to acknowledge my colleagues from the Network General days who inspired my love for protocol analysis, including Naheed Ferguson, Anita Lenk, Barbara Sandacz, Margo Lindenmayer, Don Prefontaine, Todd Perkins, Merilee Ford, Larry Young, Alex Cannara, Sean Finn, Tim Blackburn, Michelle Coomes, and many more.

And finally, thanks to Alan Oppenheimer, who is not only a protocol expert but also a terrific husband.

Joseph Bardwell: My own journey through the jungle of protocol-level communication has been a wonderful experience. Larry Denburg of the University of Delaware opened my eyes to the world of comparative programming languages in the 1970s, and many Saturday mornings found me standing at the counter of the local Radio Shack store programming the TRS Model I computer. I'm pleased to see that my son, Joshua, has followed in my footsteps as a protocol analysis engineer and teacher. With good fortune, this book will serve as a reference for him as well as for the rest of the

networking community. I echo Priscilla's acknowledgment of our colleagues from the early days at Network General, and a very special thanks goes out to Naheed Ferguson for keeping me focused in the midst of confusion. Finally, I would like to thank my associates at WildPackets, Inc., who have worked diligently to create the EtherPeek NX and AiroPeek NX analyzers. There's a little bit of all of us in that software, and that means there's a little bit of me in there too.

Introduction

The turn of the century brings us into an era of exciting advancements in computer applications. New and innovative ways of integrating computer network technology into business, education, government, and even private homes have shifted focus from the underlying engineering that allows computer systems to talk to each other. And yet, when the systems don't talk to each other, the first thing people say is, "The network is having problems!" This book discusses methods for addressing those problems and explains how networks really work. Cisco and industry-standard troubleshooting methods for analyzing, diagnosing, and fixing problems are described in detail. The book also covers techniques for using protocol analyzers, such as the WildPackets EtherPeek and AiroPeek products, to recognize and isolate faulty network behavior.

Why We Wrote This Book

We wrote this book to provide technical people with technical information that they can apply to production environments and day-to-day network configuration, support, and troubleshooting. During the 1980s and 1990s, we worked with many experts in the computer industry, some very closely, some only in passing. It became clear who the experts were, because they all knew how networks really function. None of these people said "TCIP," and they all knew that a bridge operates at Layer 2 and a router at Layer 3. They also knew lots of other things.

Many people in the computer industry were not experts, however. They thought they understood many things, but they lacked certain fundamental knowledge. Lacking formal computer network education, and forced to use, implement, support, and maintain complex systems, they drew many erroneous conclusions and sometimes taught these to their peers.

This book focuses on many of the technology and engineering issues that are often misunderstood. In reading these pages you may encounter concepts that seem to contradict what others have told you. We have attempted to put down on paper some of the core information that is critical to successful troubleshooting and protocol analysis. This information is based on documents from renowned standards organizations such as the Internet Engineering Task Force (IETF) and the Institute of Electrical and Electronics Engineers (IEEE). Of more importance, the information is based on many years of analyzing real-world, diverse, and complex networks. This book has a unique protocol-level focus that is not found in most of the volumes of technical literature available today.

Guaranteed Not to Rust, Bust, or Collect Dust

Computer networks are like used cars, and, just as when you go to a used-car lot, you have to be careful not to get a lemon. The following can be said of both networks and used cars:

- They can be made to look good when you first examine them, but they sometimes have parts that are ready to fail when you need them most.
- An inexperienced technician can tinker with them and patch up the obvious problems, leading to the incorrect conclusion that the technician is an expert.
- Some problems have deceptive symptoms and only a true expert can discern the real causes.
- Even though the basic systems and technologies that make them work have not changed much over the years, they both incorporate whatever engineering schemes were popular at the time they were designed.

We were taught to pump the brakes if a car skids on an icy road, but this rule doesn't apply when using anti-lock breaking. We were taught to turn into a skid, but this doesn't apply when the car has front-wheel drive. Changes in automotive technology completely change the way we think about some of the fundamental aspects of driving.

We were taught that there are three Internet Protocol (IP) address classes (A, B, and C) that are used for unique host identification, but this doesn't apply when classless addressing is implemented. We often hear that network utilization shouldn't exceed 40 percent on Ethernet networks. However, this is no longer true on full-duplex Ethernet links. Changes in computer network technology change the way we think about fundamental methods of design and troubleshooting.

Whether you're inspecting a used car or troubleshooting a computer network, you have to be on guard for hidden problems, advice from inexperienced helpers, confusing symptoms, and the fact that design evolution brings with it changes in terminology and function.

Audience and Scope

The audience for this book is network engineers, administrators, and technicians who manage Cisco and multivendor campus networks. A campus network is a network that spans buildings and consists of wired and wireless technologies that connect clients and servers. Although the word *campus* often refers to colleges or universities, and this book is perfectly matched to the needs of college network administrators, the book is not just for college network administrators but for any administrator who manages a campus network based on the following technologies:

- 10-, 100-, and 1000-Mbps Ethernet connectivity.
- 802.11 wireless communication.
- Switched connections between machines within a single network.
- Virtual Local Area Networks (VLANs) that segregate networks in a mesh topology.
- Routed connections between networks in a campus environment.
- Wide Area Network (WAN) connections between campus networks. (Although this book focuses on Local Area Networks [LANs], WAN information is also provided.)
- Upper-layer protocols from the Transmission Control Protocol (TCP)/Internet Protocol (IP), AppleTalk, Novell NetWare, and Windows networking protocol families.

This book isn't about figuring out if a cable is disconnected in a simple LAN; rather, it is about troubleshooting complex internetworks with tens, hundreds, or even thousands of users. This book is for network engineers who manage and configure internetworking devices. Although it doesn't cover workstation or server configuration, some of the information in this book will help desktop support personnel and server administrators also.

Finally, this book is also written for certification candidates, in particular, candidates for Cisco certifications and the vendor-neutral Network Analysis Expert (NAX) certification program sponsored by WildPackets Academy.

Cisco Certifications

In the Cisco arena, this book focuses on the Cisco Certified Network Professional (CCNP) and the Cisco Certified Internetwork Expert (CCIE) certifications.

The CCNP certification indicates advanced or journeyman knowledge of networks. Having the CCNP certification denotes to employers that you can install, configure, operate, and troubleshoot multiprotocol LAN, WAN, and dial-access services for organizations with networks from 100 to more than 500 nodes. To achieve CCNP status, you must pass five tests. This book focuses on the most advanced test, which is the Support Test. All the topics in Cisco's list of topics for the Support Test are covered.

This book is also for CCIE candidates. To achieve CCIE status, you must pass both a qualification written exam and a hands-on lab exam. This book will help you with the following CCIE Routing and Switching Qualification Exam topics:

- Cisco device operation
- General networking theory
- LAN addressing
- 10-, 100-, and 1000-Mbps Ethernet encapsulation, media access control, topologies, errors, and limitations
- Logical Link Control (LLC) 802.2
- Bridging and LAN switching
- TCP/IP
- IP routing protocols
- Desktop protocols including Novell NetWare and Windows networking
- Performance management
- WAN addressing, signaling, and framing

Because this book focuses on troubleshooting, it will also prepare you for the CCIE lab test. Now that Cisco has moved from a two-day lab test to a one-day lab test, applying efficient troubleshooting methods is even more important than it once was. The methods taught in this book will help you isolate and fix problems that appear in your lab network as you perform the difficult tasks required of the CCIE lab test-taker.

Please see www.cisco.com/warp/public/10/wwtraining for more information about Cisco certification programs.

The NAX Certification Program

The NAX certification program is an industry-standard, vendor-neutral program sponsored by the WildPackets Academy. Since 1990, WildPackets has been developing user-friendly and affordable tools for designing, maintaining, troubleshooting, and optimizing computer networks. WildPackets products include EtherPeek for Ethernet network analysis and AiroPeek for 802.11 wireless network analysis. Both of these products include NetSense real-time expert system technology for automated problem analysis. The TokenPeek analyzer addresses the needs of 802.5 Token Ring users.

To pass the NAX certification tests, a candidate can use WildPackets or other industry-recognized protocol analyzers. The candidate downloads an analyzer trace file and answers questions about real-world network problems. The exams test a candidate's understanding of protocols and ability to apply protocol analysis techniques to typical network problems. Achieving NAX certification involves three steps:

1. The Applied Analysis Technician (AATech) certification.
2. The Protocol Analyzer Specialist (PAS) certification.
3. The NAX certification.

These certifications require passing knowledge exams and practical skills exams. The knowledge exams require a candidate to demonstrate solid understanding of protocol analysis concepts and detailed knowledge of the Open System Interconnection (OSI) Reference Model and the protocols that operate at the various layers of the model. The practical skills tests require a candidate to demonstrate proficiency with a protocol analyzer. To achieve NAX certification, a candidate must also write a dissertation (white paper) on a topic selected from a list of topics approved by WildPackets Academy. This book will help with all of the knowledge exams in the NAX certification program as well as provide a solid foundation for the protocol-related aspects of many other industry certifications.

Please go to www.nax2000.com and download the Pre-Test Study Guide and Test-Taking Instructions document for complete details on the NAX certification program.

Organization

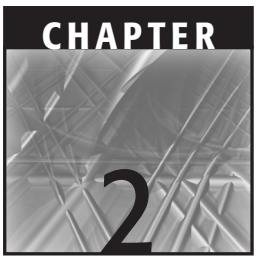
This book is organized in a bottom-up fashion. After an essential chapter on troubleshooting methods, the book works its way up the OSI Reference Model, starting with physical and data link layer concerns and ending with upper-layer concerns. The chapters are grouped as follows:

- Chapter 2 covers methods and tools for problem isolation, including Cisco and industry-standard troubleshooting procedures and protocol analysis with WildPackets or other analyzers. Chapter 2 also covers the OSI Reference Model and the Internet Control Message Protocol (ICMP).
- Chapters 3 and 4 explain how Ethernet and 802.11 wireless networks work and how to troubleshoot them when they don't work. Chapter 3 also addresses 802.2 LLC.
- Chapters 5 and 6 remain at the data link layer and address the Spanning Tree Protocol, which is used on bridged and switched networks, and the configuration and troubleshooting of VLANs.
- Chapters 7 through 8 move up to the network layer and beyond, and cover IP addressing, IP routing protocols, a detailed analysis of TCP, and an overview of upper-layer TCP/IP protocols.
- Chapters 9 through 12 teach troubleshooting and protocol analysis for the most popular desktop protocols—Novell NetWare, AppleTalk, and Windows networking.

- Chapter 13 discusses WAN technology and troubleshooting from the perspective of the LAN-oriented network engineer.

Our Web Site

We have set up a Web site, which we hope you will visit often. The Web site will include updates as new information about troubleshooting becomes available. It also includes links to practice tests to help you study for certification exams, and suggestions for exercises you can try in a lab network to strengthen your troubleshooting skills. The address of the companion Web site is www.troubleshootingnetworks.com.



Troubleshooting Methods

Computer networking isn't new. Organizations have connected computers for many years to distribute information, exchange messages, back up data, and share peripherals, such as printers and modems. Compared to those early networks, however, modern networks are complex and indispensable. Applications today include controlling space-station robots, providing medical images to surgeons, selling products, and managing manufacturing resources. Users depend on their networks for daily operations, research and development, and strategic planning. As a network support engineer, you are probably aware of increasing requirements for application support and reliability. This chapter teaches practical methods you can use to troubleshoot network problems, regardless of the newness of applications or the high level of service expectations.

Challenges in Today's Networking Environment

In the early 1980s, a large Personal Computer (PC)-based network consisted of possibly 20 workstations connected so users could share a printer and high-capacity hard disk (possibly as large as 5 Megabytes!) Typically, the network administrator was someone who had the intelligence and motivation to tinker with desktop computers. A staff of dedicated support engineers responsible for maintaining the network was little more than fantasy.

By the early 1990s, networks had changed dramatically. Twisted-pair cabling replaced the old coaxial Ethernet cabling. Switches started replacing hubs, and all large networks had dedicated teams of support people. The single file server with a handful of users sharing a disk drive evolved to a campus network that connected departments and buildings into a multiprotocol and multivendor resource critical to meeting productivity goals. However, global connectivity and the integration of voice, video, and data were little more than fantasy.

With the creation of the World Wide Web in the mid-1990s, a new epoch of networking was entered. Connectivity options for geographically separated campus networks expanded and large internetworks that provided access to employees, business partners, and customers became common. The final few years of the decade were focused on solving the Y2K problem. While the uninitiated smiled smugly on January 1, 2000, when almost none of the prophesied Y2K problems actually happened, the many unsung heroes of the software development community and the exhausted network support engineers knew that it was their hard work that caused the prophets of doom to be wrong.

Y2K came and went. With fervor, the networking industry redoubled its push to evolve. Today, global Wide Area Networks (WANs) connect campus networks composed of Ethernet and wireless devices, real and Virtual Local Area Networks (VLANs), switches and routers, Windows and UNIX servers, and a myriad of other technologies, both old and new, that must work together seamlessly and reliably. Those of us responsible for designing, implementing, supporting, troubleshooting, and repairing contemporary networks are faced with challenges on many levels. We must keep pace with accelerating technological and business changes while maintaining and troubleshooting a mission-critical production environment that was built on technologies that were developed 20 or 30 years ago.

The core engineering technologies that were invented in the 1960s and 1970s are still with us. Ethernet Version II, which was a direct evolution from the 1970s creation of Ethernet Version I, was standardized in 1982. Ethernet II frames remain typical for most Internet Protocol (IP) campus network traffic today. IP itself has not changed much since it was first created more than 20 years ago. Of course it has been repaired, and new services such as the Dynamic Host Configuration Protocol (DHCP) have augmented its functionality. But the issues of aggregating and segregating machines into reachable locations challenge the campus network designer, engineer, and administrator in much the same way as when the protocol was first created. Ethernet and IP are two of many implementations that have been with us for 15, 20, or 30 years. Today's network expert must have a solid understanding of the core engineering technologies that underlie contemporary networking.

Essentially all campus networks today implement the TCP/IP protocol, but many large networks also include AppleTalk and Novell NetWare in the mix. In business and educational environments, the desktop computer is typically based on Microsoft Windows or Mac OS, and the UNIX/Linux proponents will just have to keep on promoting the benefits of their solutions while the Bill Gates juggernaut drives the industry. Servers, on the other hand, are based on both Microsoft Windows and UNIX/Linux as well as Mac OS.

Putting all of these pieces together requires the development of a solid, effective troubleshooting methodology. It's not enough to simply "swap till you drop" and hope the problem goes away. (*Swap till you drop* is a phrase often used in the United States to mean a method of troubleshooting that involves replacing network components until you get so tired that you drop to the ground.) The significance of a good methodology will become even more evident as the present gives way to a faster, increasingly automated, and more sophisticated future.

Using a Systematic Troubleshooting Method

Network engineers who are experts in their field use a systematic process when troubleshooting problems. These experts may not be able to explain the process, but they can intuitively apply it when problems occur. Other experts evangelize their systematic methods, sometimes to the point of irrational criticism of different methods that work just as well. All the experts' methods, whether articulated or not, boil down to a logical set of steps for eliminating the causes of a problem. Experts use methods that are rational, efficient, practical, and applicable to most situations. They document the network architecture and device configurations before, during, and after making changes. They test the changes they make while troubleshooting to ensure the problem is fixed and no new problems have been introduced.

A systematic troubleshooting method is measurably different from the methods used by novices in the networking field. Novices often use random methods with no planning, no documentation, and no testing. Their methods often fail to find the actual problem in the quick time frame demanded by network users, and their efforts often leave the condition changed but still broken. An expert, on the other hand, can narrow down possibilities quickly, isolate the problem, fix the problem, and test the fix in a reasonably short time.

An expert can either intuitively or consciously define the following terms and use them to increase efficiency. These definitions are based on definitions in the online version of *Merriam-Webster's Collegiate Dictionary*:

Component. A constituent part or ingredient of a whole.

Function. The action for which a thing is specially fitted or used.

System. A regularly interacting or interdependent group of components forming a unified whole.

Troubleshoot. To locate trouble and make repairs in machinery and technical equipment; to anticipate and solve problems.

Upgrade. To raise the quality of.

Systems, including network systems, are made up of interrelated components. Systems and components can be described by what functions they carry out. Troubleshooting and upgrading a system involves considering the components of the system, their functions, and how they affect the whole system. Malfunction of any

component may cause the whole system to fail. Many newcomers to the networking field have not considered these basic concepts. Without any understanding of systems, components, and their functions, a newcomer has a difficult time troubleshooting. Going to the next step—from thinking about systems to using systematic methods—is also difficult for novices, especially in the current environment where networks have become quite complex and the interrelation of network components is not as clear as it once was.

One factor that distinguishes an expert from a novice is that the expert has researched how protocols work. Using a protocol analyzer, the expert has studied frame formats, protocol behavior, connection establishment, reliability mechanisms, routing protocols, and so on. It is not possible to learn all protocols and how they fit together, however. Protocol design engineers introduce new protocols and new ways of using old protocols on a regular basis. By the time a network engineer figures out one protocol, there's a new protocol, or, a new way of using an old protocol, to confuse the issue. Considering network behavior and troubleshooting from a higher level is a necessity in the crazy mix-and-match world of today's network protocols.

Using the OSI Model for Troubleshooting

An effective method for troubleshooting and understanding protocols is to use the International Organization for Standardization (ISO) Open System Interconnection (OSI) Reference Model as a guide. Figure 2.1 shows the OSI model.

The OSI model has seven layers. Each layer communicates with its peer running on another system. The layer doesn't do this directly, however. With the exception of the bottom layer, a layer passes its data to the layer below for further processing. Each layer provides services to the layer above. Table 2.1 describes the services provided by each layer of the model.

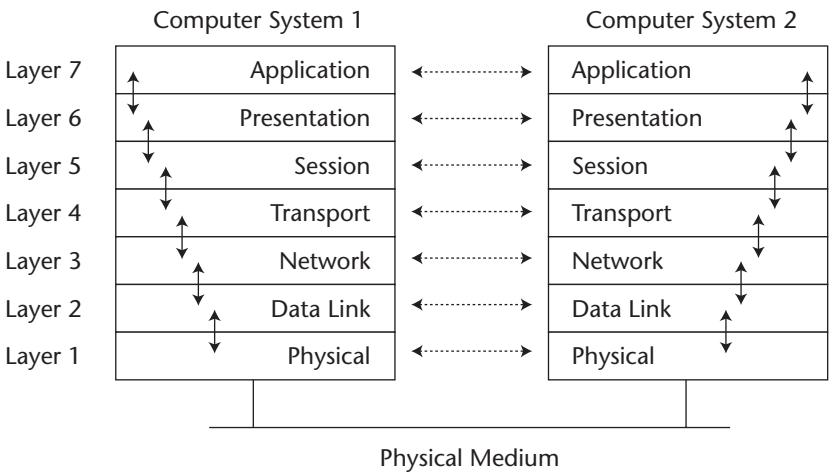


Figure 2.1 The OSI model.

Table 2.1 Services Provided by the OSI Layers

LAYER	SERVICES
Application	Handles file and message transfer, directory lookups, naming, authentication, and other services required by applications
Presentation	Ensures that information sent by one system will be readable by the application layer of another system
Session	Establishes, manages, and terminates sessions between applications
Transport	Provides end-to-end communication, error recovery, and flow control
Network	Provides connectivity and path selection between networks
Data Link	Provides transit of data across a physical link
Physical	Defines the electrical and mechanical specifications for physical links between systems

It's important to understand the services offered by each layer of the OSI model and typical problems at each layer. When isolating the cause of a problem, you should work your way up from the bottom to the top layer. Starting at the bottom layer, check cabling and physical interfaces first. Next check encapsulation options at the data link layer and any problems with media access control. Move up the layers to routing protocol problems and network-layer addressing issues. At the upper layers, check for problems with buffer exhaustion, software bugs, naming, encryption, compression, data presentation, and user errors. The rest of the chapters provide more detail on these suggestions.

Newcomers to the field of networking faithfully memorize the OSI model and its functions. As the newcomers learn more about networking, they learn that many protocols were designed without reference to the model and do not fit the model perfectly. Some protocols do not fit into a linear stack at all, but rather sit outside the main list in a management or control plane. Despite these caveats, the model does a good job of explaining the typical functions required of a networking protocol and makes a good guidepost when troubleshooting.

The OSI model provides a skeleton for understanding both existing and future protocol behavior. When experts see the progression of addressing in a packet (from data link, through network, to the transport layer), they can apply an understanding of the OSI model to extract the meaning and function of the packet. Associating network devices with OSI layers (switch at Layer 2, router at Layer 3, file server at Layer 4 and above), engineers can correlate network symptoms with potential sources of network problems. Symptoms at the network layer may indicate a misconfigured routing protocol on a router. Retransmissions at the transport layer may indicate a problem with a shortage of memory or computing power on a file server. This kind of thinking can be extrapolated for numerous networking problems.

ANOTHER CAR ANALOGY

An expert protocol analyst is similar to an expert auto mechanic. You and I know that the gas goes in a hole on the outside of the car. Then the gas sort of mixes up in the cylinders in the motor, where the spark plugs explode the gas, and that makes the motor turn. Then the transmission connects the motor to the wheels, and so on. When we hear a funny noise we say, "Hmm...that sounds like I'm running out of gas," or "Sounds like I have a flat tire." We're not expert auto mechanics.

Trained mechanics, however, understand the nuances and relationships of the car's fuel system, ignition system, drive train, brakes, electrical system, and so forth. Although different makes of cars are different in many ways, mechanics can probably figure out many systems of cars they may not normally work on. So, too, with the OSI Reference Model and networking experts. Experts understand flow control, message encoding, acknowledgments, routing, and so on, whether they are working with Ethernet, Asynchronous Transfer Mode (ATM), TCP/IP, or some new protocol just out of the Internet Engineering Task Force (IETF).

The OSI model helps network engineers understand that protocol design is modular. Each layer of the model operates semi-independently and offers a service interface for a layer above it. When the OSI concept of independent layers isn't followed, problems arise. Take the case of the File Transfer Protocol (FTP), which has a 32-bit IP address encoded into the application layer, which causes challenges for Network Address Translation (NAT). Chapters 7 and 9 discuss NAT and FTP in more detail.

Because the OSI layers are semi-independent, protocols can be stacked in creative ways. For example, many companies route Systems Network Architecture (SNA) traffic over an IP network. The result is that the SNA traffic, which previously resided only on a Token Ring network, is now encapsulated in Logical Link Control (LLC) and Token Ring headers, encapsulated again in TCP/IP headers, and finally encapsulated again in another data link layer for traversal across the IP internetwork. An expert who understands the OSI model is not surprised to encounter this sort of creative layering when troubleshooting protocol behavior.

Although some network specialists scoff at the idea of using the OSI model for troubleshooting, anyone who claims to be an expert at troubleshooting and protocol analysis understands and uses the model. This book uses the model extensively and will help you, the reader, use the model to become an expert protocol analyst.

Generic Problem-Solving Models

Generic problem-solving models are a topic for research in the engineering, information science, business, and psychology fields. A generic model allows you to solve unstructured and complex problems of any sort. The study of problem-solving methods is important for both practical and theoretical reasons. Problems do not only occur in the technological realm. Learning problem-solving skills helps a person cope psychologically and cognitively.

Many high school and college classes, including the Cisco Networking Academy classes, teach the generic Dartmouth Problem-Solving and Design Method, developed by the Thayer School of Engineering at Dartmouth College. The Dartmouth model is definitely applicable to network troubleshooting. The Dartmouth approach solves problems by proceeding through a problem-solving cycle, step by step, carefully documenting each step in the process. If users of the model discover that a solution they are working on is not viable, they examine their paper trail and move back only as far as necessary—perhaps only a single step. The steps in the Dartmouth method are as follows:

1. State the problem after examining it carefully.
2. Redefine the problem to eliminate any bias of the customer reporting the problem or any preconceived notions about the ideal solution.
3. Identify any constraints on potential solutions and outline general specifications and goals for the solution.
4. Identify alternative solutions using brainstorming techniques that are structured by the constraints and goals identified in the previous step.
5. Analyze the alternatives, weighing the advantages and disadvantages of each.
6. Select and test the most viable alternative.
7. Iterate the cycle until you find the right solution.

The Dartmouth model is one of many generic models for problem solving. Generic models share important characteristics. All require the user to define and redefine the problem and to analyze and plan possible solutions. The generic models include a set of steps that are applied iteratively. The models are heuristic and incorporate knowledge gained from the testing of real-world conditions into the next step. The models are not random. They are not similar to the typical methods used by networking novices, which have just one iterated step: Make changes, without documenting them, until the problem seems to go away.

Fundamental Network Troubleshooting Considerations

Whatever troubleshooting method or model a networking professional develops, there are two fundamental questions that must be answered in every case:

What is the problem that needs to be solved? Without carefully considering what is actually wrong, there is no way to know whether a particular piece of evidence relates to a solution. Typically, a complex network has many inefficiencies and possibly quite a few minor problems. The complaint that stimulated a call to action must be quantified completely or else initial research into the cause of the problem may turn up other unrelated problems. Fixing the unrelated problems won't address the complaint.

What is the impact of the problem on the end-user community? Troubleshooting tools may identify numerous network anomalies. You need to ask yourself whether these anomalies are really affecting user productivity and satisfaction.

For example, if a large file transfer takes 130 seconds on Monday and 140 seconds on Friday, it is unlikely that any user will notice the difference. If a user is working on the Web and the response time goes from 4 seconds to 14 seconds, then there's going to be a complaint. Simply tabulating various network statistics has little meaning unless the statistics are correlated to the activities of the end-user community. This implies that the person doing the troubleshooting must have a reasonable understanding of the needs and expectations of the users of the computing system.

Cisco's Troubleshooting Method

For the Cisco Internetwork Troubleshooting (CIT) class, Cisco developed a troubleshooting method based on generic problem-solving models. In the CIT course materials, Cisco acknowledges that many systematic methods work for troubleshooting computer networks. Engineers are not expected to use the Cisco model simply because they use Cisco equipment. To pass the Cisco Support Test, however, you should learn the troubleshooting method verbatim. One purpose of this book is to help you pass Cisco tests, so this section describes the Cisco troubleshooting method, which is shown in Figure 2.2.

The Cisco troubleshooting method is a practical process that shares characteristics with any systematic method. The initial goal is that you have a clear and sufficient definition of the problem. After defining the problem, you should gather more facts and consider possible causes for the problem. Next, create an action plan for how you will test your theories about possible causes. Then, implement the plan and observe the results. If the symptoms don't stop, develop and try another action plan. It may be necessary to gather more facts at this point. If the symptoms stop, document how you resolved the problem. The following sections describe these steps in more detail.

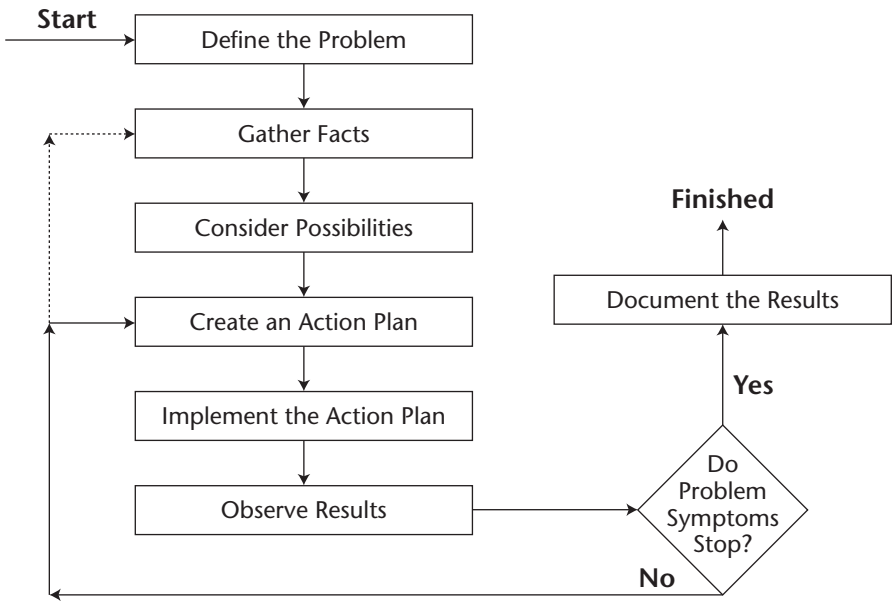


Figure 2.2 Cisco's troubleshooting method.

THE IMPORTANCE OF DEFINING THE PROBLEM TO BE SOLVED

A large metal parts fabrication company on the East Coast of the United States hired one of the authors as a protocol analyst to help with complaints of slow network performance. While walking around with the network administrator, the consultant noticed that users were literally leaning back in their chairs waiting for the company's database server to respond to their queries. The response time was in the neighborhood of 20 to 30 seconds! Back in the network administrator's office the consultant pointed out the evidence of slow response time that had been observed. The network administrator said, "Oh, I know, but please don't say anything to any of the users. They think that's normal and nobody is complaining. We're budgeted to upgrade the server next quarter. The problem I need help with is that our users in remote offices are getting over 60-second delays using the database." Had the consultant not stumbled onto a quantification of the complaint, there could have been many wasted hours tracking down the wrong problem.

Define the Problem

The first step is to define the problem. Writing a description of the problem and its symptoms is often a valuable step that will save time later. Writing may seem like a bureaucratic requirement, but it has practical value in that it helps you clarify the situation and requires you to analyze symptoms and possible causes. In some cases, writing about the problem causes you to realize the solution immediately, greatly reducing the amount of work in the following steps. When writing about the problem, form your observations with reference to the baseline you established for your network. You should have a good idea of what is normal for your network in terms of errors, throughput, response time, and efficiency. Describe to what extent the current situation deviates from normal.

In this first step, focus on symptoms and what might cause them. Many ideas may arise, but concentrate on those that could be major contributors to the problem. Focus on users' reports of the problem, but, at the same time, be somewhat skeptical of user reports. Users don't always tell the whole truth about the problem because they are afraid of looking dumb. Also, as mentioned in the *Generic Problem-Solving Models* section, users sometimes incorporate bias and preconceived notions about the solution into their descriptions of the problem. Proceed from your own knowledge of the problem and your understanding of the internetwork topology and typical network performance.

Gather Facts

Gather additional facts from affected users, network administrators, managers, and any key people involved with the network. Collect data from network management systems, protocol analyzers, router and switch diagnostic commands, software release notes, software bug reports, and documentation about changes made to the network. Check records that you kept (hopefully!) on the configuration of hosts, servers, switches, routers, and any other configurable network devices. Compare current configurations with saved configurations to see if anything has changed. Determine answers to the following questions:

- How often does the problem happen?
- When did the problem first occur?
- What changes were made right before the problem started happening?
- Is the problem reproducible?

Consider Possibilities

Using the data you gathered and previous knowledge you gained from proactive network monitoring, set some boundaries to help you effectively isolate causes for the problem. Consider probable possibilities and set aside details that seem irrelevant. Analyze symptoms to determine which possibilities are unlikely. For example, if local stations can communicate with each other but not with remote stations, you know that the Network Interface Cards (NICs) in the stations are operational. A likely possibility is that the problem lies above the physical and data link layers. Document possibilities and list them in order of most likely to least likely.

Create an Action Plan

Develop a plan for how you will test the most likely causes of the problem. Plan to change just one variable at a time; otherwise you won't know which change resulted in the fix. It often helps to use a *divide-and-conquer* approach whereby you partition your troubleshooting domain into discrete areas that are logically or physically isolated from each other. This approach will help you pinpoint the cause(s) of the problem. A *testing-outward* approach also helps in many cases. For example, from a source station, plan to test local communications first. Then create a plan to test reachability to each network along a path to a remote network until the problem occurs.

Document your action plans. Each plan should describe a set of steps to be executed. In addition, it is important to have a backout plan in case your actions make matters worse. For example, in your action plan, include an initial step of saving existing configurations. If problems occur, you can go back to a known state with the saved configurations.

Implement the Action Plan and Observe the Results

Follow the steps that you created in your action plan and observe the results. Make sure you document which plan you are currently trying; otherwise it is too easy to repeat yourself. Test all fixes that you make. Be sure you do not make the problem worse or introduce new problems. Try to limit as much as possible the invasive impact of your actions on network users. Also, minimize the extent and duration of any security lapses that could occur while implementing your action plan. If necessary, execute the backout plan if it appears that your actions are detrimental to security or network operations.

After manipulating a single variable based on your action plan, gather data to determine whether your action has fixed the problem. Ask users to try their applications

again and to let you know if the problem is cured. If you determine that you have solved the problem, document the results and any changes you made. If you have not yet fixed the problem, repeat the problem-solving process. Implement another action plan, and, if necessary, gather more facts.

Document the Results

When you have resolved the problem, you have one more important step remaining—documenting the results. You may have tried many action plans. Be sure to complete the job by recording which plan worked and why (if you know why). Documenting the resolution will help you in the future when a similar problem occurs. Also, if you determine later that your fix unexpectedly caused another problem, you will have a paper trail of what has changed. In addition to documenting the resolution, be sure to save any configuration changes you made. If necessary, update your network map(s).

NOTE **In the Cisco Certified Internetwork Expert (CCIE) lab test, the proctor will check that you document your work.**

Proactive Troubleshooting and Baselining

How can you know something is wrong if you haven't defined the opposite of wrong? A *baseline* of your network defines normal performance and typical protocol behavior when no problems are occurring. With a baseline of your network, you can more efficiently troubleshoot problems. Instead of simply relying on user reports, you can check statistics and configurations to see what is different from the baseline. You can capture data with a protocol analyzer and compare the results to data you captured before. With a baseline, you have a basis for comparison so that problems can be more easily identified.

The online version of *Merriam-Webster's Collegiate Dictionary* defines a troubleshooter as "a person skilled at solving or anticipating problems or difficulties." According to this definition a troubleshooter must be able to *anticipate* problems. In the computer networking world, you may have encountered those eccentric, experienced experts who can seemingly foresee problems and their causes. They can immediately recognize when a problem is about to happen and often correctly guess the cause of the problem. They are like a mother who can recognize her child's cry in a nursery room filled with crying children and can accurately guess why her child is crying.

Experts are aware of the nuances of the networks they manage, including the networks' frailties and nonoptimal portions, and they can anticipate when something is going to break. This isn't magic. The experts can do this because they have worked on the network for a long time and because they spend time proactively monitoring it. Instead of using troubleshooting tools only to react to problems as they occur, experts use the tools to proactively develop a baseline for normal network behavior and performance. When problems occur, these experts can compare current problems with the baseline.

THE IMPORTANCE OF NETWORK DOCUMENTATION

One of the authors was hired by a major aerospace company to do protocol analysis. After examining network statistics and trace files, the author asked the network manager to check some configurations on the FS01 file server. The network manager replied, "Oh, FS01....Yeah, we know that one's acting up, but we don't know where it is anymore." The FS01 file server was somewhere on the company's campus—probably in some wiring closet—but nobody remembered where it was located!

Documenting Your Network

One goal of proactive network management is to document your network's logical and physical topology. Documenting a complex, ever changing network is challenging, but the benefits of having detailed network topology and configuration information make the effort worthwhile. Learning and documenting the locations of major hosts and servers, interconnection devices, and network segments is a good way to develop a basis for future troubleshooting and optimization. Coupled with data on the performance characteristics of network segments, location information gives you insight into where users are concentrated and the level of traffic a network must support. Network maps are important for both proactive and reactive network management. If a protocol analyzer or other management tool identifies that a device—for example, a file server—is misbehaving, you need to find that server. The map should help with this task.

Network managers often resist documenting their networks because they are so busy handling day-to-day operations and planning enhancements to the network. To overcome this resistance, consider the fact that good documentation is directly related to money. Your company has probably invested in many troubleshooting and network management tools. Perhaps you have an umbrella management system, such as Cisco Works, or protocol analysis equipment, such as WildPackets EtherPeek or AiroPeek. Your network may implement Remote Monitoring (RMON) probes and use a centralized RMON or Simple Network Management Protocol (SNMP) console to alert you to problems and provide ongoing statistical reporting. These are all useful tools, but don't forget that these tools identify numeric Ethernet, IP, AppleTalk, NetWare, DECnet, and other types of addresses. If you have no idea of the actual physical location of these addresses, then your troubleshooting capabilities are impaired and the investment you made in all the clever tools may be wasted.

Your goal should be to develop a map or set of maps that include the following information:

- Geographical information, such as a country, state, province, city, or campus name
- Buildings and floors, and possibly conference rooms, offices, and cubicles
- WAN and LAN connections between buildings
- An indication of the data link layer technology for WANs and LANs
- The name of the service provider for WANs
- Circuit IDs for WANs

- Network-layer addresses and names for LANs, WANs, and major devices
- The location of the Main Distribution Frame (MDF) and Intermediate Distribution Frame (IDF) wiring closets
- The location of routers and switches, though not necessarily every hub
- The location and extent of VLANs
- The location and extent of Virtual Private Networks (VPNs)
- The location of remote-access servers
- The location of major file, print, database, application, and Web servers
- The location of mainframes
- The location of major Network Management Stations (NMSs)
- The location of firewalls or other security management systems
- The location and topology of demilitarized zones for computers outside the firewall
- Some indication of where workstations reside, though not necessarily the explicit location of each workstation
- A depiction of the logical topology or architecture of the network

Networking experts use both logical topological diagrams and physical topological diagrams. *Logical topologies* refer to broad categories of devices, logical interconnections, and the flow of information in a network. *Physical topologies* refer to actual devices, ports, cables, connections, and the physical layout of a network. Both types of diagrams are helpful.

When documenting the network infrastructure, take a step back from the diagrams you develop and try to characterize the logical topology of the network as well as the physical components. The logical topology illustrates the architecture of the network, which can be hierarchical, flat, structured, unstructured, layered, bus, ring, star, and so on. The logical topology can affect your ability to upgrade a network. For example, a flat topology of devices all in the same subnet connected with switches and hubs does not scale as well as a hierarchical topology that makes use of routers.

In addition to documenting network topologies, many network documentation tools enable you to save detailed information about network resources. Documentation tools often include a device object library that contains detailed information about network objects, such as their configuration, serial number, make and model, software and hardware version, software license number, and so on. Of utmost importance is the name and network-layer address for the object, as discussed in the next section.

Documenting Network Names and Addresses

When drawing detailed network maps, you should include the names of major sites, routers, switches, network segments, and servers. Also document any standards for naming network elements. For example, some network engineers name sites using airport codes (Atlanta = ATL, Boston = BOS, and so on). Some engineers suffix names with an alias that describes the type of device (for example, rtr for router).

You should also document network-layer addresses and any standard policies for assigning addresses. Addressing information will be extremely valuable when looking at protocol analyzer packet data when troubleshooting. An understanding of the addressing policies will help you know which areas may have problems when you upgrade or optimize the network. The addressing policy (or lack of any policy) can make it difficult to upgrade. For example, current IP subnet masking may limit the number of nodes in a LAN or VLAN. Poorly designed addressing architectures might limit the use of *route summarization*. Route summarization reduces the number of routes in routing tables and minimizes routing table update traffic and overall router overhead. Route summarization also improves network stability and availability, because problems in one part of a network are less likely to affect the whole internetwork. Summarization works best if addresses have been assigned in a consistent and contiguous manner.

Tools for Network Documentation

To develop good network documentation, you should invest in good network diagramming tools. Although some smaller campus networks are documented with generic drawing tools such as CorelDRAW and PowerPoint, for larger campus networks you should use a tool that includes icons for typical devices, an object library for detailed data, and support for *autodiscovery*. With autodiscovery, a tool can learn about devices and topologies automatically by listening to traffic and sending queries and analyzing the results. Good diagramming tools also support printing large network diagrams on plotters. They also support conversion to the HTML and GIF formats for posting the data on a Web server so other network engineers and users can share the data.

CAUTION **If you store your network maps on a server to which many engineers have write access, be sure to have a change control process in place. When making a change, an engineer should check the map out. Other engineers should be unable to make a change until the first engineer checks the map in again.**

The Microsoft Visio product line is highly recommended for network diagramming. The product line includes Visio Standard, Visio Professional, and Visio Enterprise Network Tools. You can check Microsoft marketing material to determine which tool is best for your needs. All Visio products share a common file format, which means you can share diagrams with other Visio users, regardless of which product you choose.

Many network engineers recommend the netViz products from netViz Corporation. netViz is an information management tool that allows you to visualize and work with complex data systems such as internetworks. Designed to deal with large amounts of information, netViz integrates graphics and data to create a visual database, making it easy for you to see system components, their unique characteristics, and their relationships to each other.

For large campus networks and service providers, Visionael Corporation offers client/server network documentation products that have network inventory, troubleshooting, and change management features. Visionael products support network

planning, design, deployment, provisioning, validation, and daily operations. Visionael products provide detailed data about the physical topology as well as the logical topology.

For smaller networks that are managed by engineers who prefer Mac OS, a popular application is InterMapper from Dartware, LLC. With InterMapper you can quickly develop maps and see the state of your network at a glance. InterMapper also provides detailed information about devices, servers, and links. Another popular product in the Macintosh community is LANSurveyor from Neon Software.

For an overall view of Internet and TCP/IP measurement and diagramming tools, check the information collected by the Cooperative Association for Internet Data Analysis (CAIDA). CAIDA categorizes tools with respect to their intent and provides a summary along with Web page pointers to more detailed information. Review comments are also included when available. For more information on the CAIDA catalog, go to www.caida.org/tools/taxonomy/.

Documenting Switched Networks

Automated network mapping tools that use autodiscovery to locate network devices may not identify bridges or switches. A Layer 2 switch or bridge is transparent at the data link layer. This means that, by design, packets that are forwarded through the switch or bridge do not carry with them any evidence of the switch or bridge being present. Unless a Layer 2 interconnect device is configured to make its presence known to an autodiscovery tool, there is no consistent way to detect its presence. It's true that Layer 2 devices typically implement the Spanning Tree Algorithm to prevent loops in the network topology, and the Spanning Tree Algorithm uses periodic transmission of Bridge Protocol Data Unit (BPDU) packets, per the IEEE 802.1D standard. All of the ports on a switch don't send BPDU packets, however, and the port that does send them may change if the network topology changes. The only way to consistently identify the presence of a Layer 2 device in a network is to configure the device to announce its presence or to configure the device to respond to queries. The Cisco Discovery Protocol (CDP), covered in the next section, is one way to configure a Cisco switch to announce its presence.

A switch can be configured with an IP address in which case the switch is reachable via management tools such as ping, SNMP, and Telnet. If the switch implements VLANs, however, you can only reach it from devices that are in the same VLAN or are across a router interface that is in the same VLAN. By default, Cisco switches place management functions, such as ping and SNMP, in VLAN 1. The management VLAN can be changed, however. You should make sure any connected router or management ports are also in the management VLAN to ensure you can reach the switch for troubleshooting and remote configuration.

In general, VLANs make network documentation more challenging. In the old routed world, managers documented the network-layer addresses of network segments and major devices such as routers and servers. In a switched VLAN world, you may need to add much more detailed, Layer 2 documentation in order to understand your network. You should document the names and numbers of each VLAN and which ports are associated with the VLANs. You should also keep track of the Media Access Control (MAC) addresses that the switches use for management functions.

Cisco switches use multiple addresses for the Dynamic Inter-Switch Link (DISL), the Virtual Trunk Protocol (VTP), BPDUs, and other management and control functions. You will learn more about VLANs in Chapter 6.

The Cisco Discovery Protocol

CDP is a simple tool for basic network documentation that is built into every Cisco switch and router. CDP specifies a method for Cisco routers and switches to send configuration information to each other on a regular basis. Analyzing CDP data can help you learn about your network's structure and configuration. You can use the `show cdp neighbors detail` command to display detailed information about neighboring routers and switches, including which protocols are enabled, network addresses for enabled protocols, the number and types of interfaces, the type of platform and its capabilities, and the version of Cisco Internetwork Operating System (IOS) software running on the neighbor.

In the following example, a network engineer used CDP on the Charlotte router to confirm her suspicions that the network map that showed Charlotte connected to the Boston router's serial 1 interface was wrong. The CDP information confirmed that the connecting interface on Boston was serial 0, not serial 1. The outgoing serial 0 port on Charlotte connects to Boston's serial 0 interface.

```
charlotte#show cdp neighbors detail
-------------------------
Device ID: Boston
Entry address(es):
  IP address: 172.16.40.1
  Novell address: 100.4afc.4afc.4afc
Platform: cisco CSC4,  Capabilities: Router
Interface: Serial0,  Port ID (outgoing port): Serial0
Holdtime : 156 sec
Version :
Cisco Internetwork Operating System Software
IOS (tm) GS Software (GS3-AK-M), Version 11.0(5), RELEASE SOFTWARE (fc1)
Copyright (c) 1986-1996 by cisco Systems, Inc.
Compiled Mon 05-Feb-96 22:05 by hochan
```

CDP frames are sent to the Cisco multicast address 01-00-0C-CC-CC-CC. CDP frames are sent every 60 seconds by default, although you can change this with the `cdp timer` command. CDP frames are sent with a *holdtime* of 180 seconds by default. Holdtime specifies the amount of time a receiving device should hold the information before discarding it. When an interface is shut down, CDP sends a frame with the holdtime set to zero. You can configure holdtime with the `cdp holdtime` command.

CDP is enabled by default on all routers and switches. To disable CDP and later reenable it, use the `no cdp run` and `cdp run` commands. These commands can be entered in global configuration mode to affect all interfaces or in interface configuration mode to affect a single interface.

If you wish to reset CDP traffic counters to zero, use the `clear cdp counters` command. The `clear cdp table` command is also helpful when you wish to delete the CDP table of information about neighbors. The most useful CDP command, however, is

`show cdp`. Arguments to this command let you see detailed or summary data, information about a particular neighbor, or information about a particular interface.

Protocol Analysis

To proactively learn about traffic on your network and to troubleshoot problems, you should use a *protocol analyzer*. A protocol analyzer records, interprets, and analyzes network traffic. It operates by placing its NIC into *promiscuous mode.* This means that the NIC accepts all packets without regard for the destination address. A protocol analyzer provides detailed information about packets and communication sessions. It decodes the various protocol layers and fields in each packet, and presents the layers and fields as readable text. When necessary, the analyzer decodes the meaning of each byte (or bit, for bit-oriented protocols). Relationships between packets may also be analyzed.

A protocol analyzer connects to a network and captures packets as they travel across the network. Throughout this book, we use terms such as *captured packets, capture results, capture rate*, and so on. We assume that you have a protocol analyzer and understand that to capture packets means to connect the analyzer to a network and configure it to start accepting packets into its buffer as the packets are read by the NIC in the analyzer.

An analyzer captures packets into a software buffer. In capture mode, *filters* can be set to save only traffic that meets certain criteria. For example, if a particular device is suspected of causing network problems, then a filter can be configured that captures all traffic to and from that device and ignores other packets. The analyzer also *time-stamps* each packet as it arrives. Timestamps can be extremely important when analyzing protocol operation and performance. They can also be helpful when determining response times by measuring the delta time between packets. Once data is captured, the analyzer allows the user to save the data as a *trace file*. The trace file can be loaded later for continued analysis.

Most analyzers show packet data in three formats: a summary, detail, and hexadecimal view. In the summary view, an analyzer shows one line for each packet, with a configurable amount of data displayed for each packet. This view can be helpful when tracking multiple-packet communication sessions. In the detail view, the analyzer shows the meaning of layers, fields, bytes, and bits for each packet. In the hexadecimal (hex) view, the analyzer shows the raw packet. Figure 2.3 shows an example of a summary view. Figure 2.4, later in this chapter, shows an example of a detail view.

EtherPeek

File Edit View Capture Send Statistics Tools Window Help

OSPFLab.pkt

Packets: 10

Packet	Source	Destination	Size	Time-Stamp	Protocol
1	IP-172.16.10.1	IP-224.0.0.10	78	14:05:54.892000	EIGRP Hello
2	IP-172.16.10.1	IP-224.0.0.5	82	14:05:56.800000	OSPF Hlo
3	IP-172.16.10.1	IP-224.0.0.10	78	14:05:59.893000	EIGRP Hello
4	00:00:0C:05:3E:80	00:00:0C:05:3E:80	64	14:06:00.416000	Loopback
5	IP-172.16.10.1	IP-224.0.0.10	78	14:06:04.893000	EIGRP Hello
6	IP-172.16.10.1	IP-224.0.0.5	82	14:06:06.801000	OSPF Hlo
7	IP-172.16.10.1	IP-224.0.0.10	78	14:06:09.894000	EIGRP Hello
8	00:00:0C:05:3E:80	00:00:0C:05:3E:80	64	14:06:10.417000	Loopback
9	00:00:0C:05:3E:80	01:00:0C:CC:CC:CC	300	14:06:13.423000	Discovery
10	IP-172.16.10.1	IP-224.0.0.10	78	14:06:14.894000	EIGRP Hello

Packets Nodes Protocols Conversations Size Summary History Log

Figure 2.3 Protocol analyzer summary view.

WildPackets Protocol Analyzers

We recommend the WildPackets protocol analyzers because they are easy to use, accurately decode hundreds of protocols, and are affordable. The WildPackets EtherPeek analyzer is available as a no-cost demo download from www.wildpackets.com. The demo version can't save files, print, or capture for extended periods, but it is fully functional and can provide a glimpse into the traffic on your network.

In the WildPackets EtherPeek and AiroPeek products, a real-time expert system analysis engine called NetSense runs during capture time and provides automated analysis of common network problems. NetSense can also provide limited modeling and simulation capabilities by allowing the user to change various parameters and pose "what-if" questions. NetSense also provides application response time and throughput analysis.

The iNetTools suite is part of EtherPeek and provides common network troubleshooting utilities from within EtherPeek. These utilities include ping, trace-route, whois, ping scan, port scan, finger, and a throughput measurement tool.

On his Ethernet Web site, Charles Spurgeon, the renowned author of *Ethernet: The Definitive Guide* and other books, maintains a list of protocol analyzers, including EtherPeek and others. (See the Web site at www.ethermanage.com/ethernet/software.html.) If you don't already have an analyzer, try one that is free or that has a free demo version. With an analyzer, you can reproduce the packet capture examples in this book and enhance your retention of the concepts discussed here.

Using a Protocol Analyzer in a Switched Network

When attached to a shared network, such as a set of cables and devices connected via hubs, an analyzer sees all traffic and can capture and decode all packets. When connected to a switched network, an analyzer sees broadcast, multicast, and flooded traffic. The analyzer also sees any traffic addressed to devices reachable via the same port to which the analyzer is attached. This may not be much traffic if the analyzer is the only device on that port, however. If other devices share the port—for example, if a hub is connected to the port—then the analyzer sees more data.

As discussed in more detail in Chapters 3 through 5, a switch forwards frames based on the destination MAC address in a frame. Switches filter frames that do not need to exit a particular port. Switches also forward broadcast and multicast frames, and frames for which the switch has not yet identified the location of the destination MAC address in the frame. Switches initially flood all traffic, but by examining the source address in each packet, the switch quickly learns which addresses are reachable from which port and then forwards frames selectively. This is a problem for protocol analysis, although a good thing for network performance.

There is a solution to this problem. Most switch vendors support *mirroring* traffic. With mirroring, a switch copies (mirrors) selected traffic to a monitor port. Enabling the mirroring feature tells a switch to copy traffic forwarded to the mirrored ports to an additional monitor port where an analyzer is attached. Some Cisco documentation calls the monitor port where the analyzer is attached the *Switched Port Analyzer* (SPAN) port. You can configure a single port as the monitor port and also configure which traffic should be sent to the monitor port. You can mirror traffic destined to an individual port or to multiple ports, or traffic for one or more VLANs.

LEARNING PROTOCOL ANALYSIS

In one sense, protocol analysis is the study of the language of network communication. Unfortunately, there's no way to grow up learning the language of protocols the way children grow up learning their native languages. One way to approach protocol analysis can be likened to an adult learning a foreign language for the first time. Of course, if the language uses foreign symbols, then the student will need to learn how to pronounce the symbols properly. English, Japanese, Greek, Arabic, and Hebrew all use different symbols to represent the sounds of words. So, too, the protocol analysis student will need to learn to recognize and "pronounce" binary and hexadecimal numbers, ASCII encoding, EBCDIC encoding, Manchester signal encoding, and even Reversible Half-ASCII in the NetBIOS arena. As for an American learning Japanese, there will be a learning curve just to get the basic symbols in mind.

In addition to teaching symbols and pronunciation, foreign language classes explain noun endings and possessives, verb tenses and voices, and whether or not adjectives precede or follow the nouns they modify. The students learn the structure and organization of the language. There is structure and organization in the protocol world as well. The OSI model, shown in Figure 2.1 on page 14, defines relationships between the components of network communication. In the same way that fluent speakers don't diagram sentences before they talk, or think about whether they're using the future perfect tense, the expert protocol analyst also knows how pieces of protocol languages fit together. This knowledge starts with a basic understanding of the layers in each protocol as outlined by the OSI model.

It's said that as people become fluent in a new language, there's a point when they can think in that language. When network professionals can think in TCP/IP or AppleTalk, they become intuitive problem solvers. The fluent speaker also becomes a fluent reader. You may have had the experience of reading a famous book in the native language of the author and gleaning nuances of meaning that were lost in a translation. When experienced protocol analysts "read" an analyzer trace file, they also glean the subtle behaviors expressed by the communicators. They can identify problems and performance degradation, and they can anticipate future events that may be showing early manifestations in the protocol interactions.

On Cisco switches that use the CatOS command line interface, use the `set span` command to specify which ports are mirrored and which port is the monitor (SPAN) port. Be careful with the syntax of this command. It is very similar to the `set spantree` command used by the Spanning Tree Protocol, which Chapter 5 discusses in more detail. Because Cisco lets you abbreviate commands, you might end up configuring SPAN when you meant to configure the Spanning Tree Protocol.

On most Cisco switches that use the Cisco IOS command line interface, the command to configure mirroring is `port monitor`. On some IOS switches, the command is `monitor-port`. You should first enter the command to enable the monitoring feature and then configure two parameters—the port where the analyzer resides and the ports that will be monitored.

NOTE **Some Cisco switches disable the forwarding of unknown (flooded) unicast traffic and unregistered multicast traffic to the monitor port. The goal is to protect the network from problems associated with this traffic exiting an additional port. For example, a Cisco Catalyst 1900 switch automatically adds the `port block unicast` and `port block multicast` commands to the monitor port. This can make troubleshooting difficult. The workaround is to upgrade to a higher-end switch.**

The following output shows a common mistake that is made with the `monitor-port` command. The user assumed that port monitoring was working, but in fact it was not working because only the parameters had been configured. The port monitoring feature had not been enabled. The user had a protocol analyzer connected to switch port 0/11. The user expected the analyzer to see the ping packets that traveled from switch port 0/12 to an external network, but the analyzer did not see the packets.

```
(config)#monitor-port ?
  monitored  Set monitored port
  port       Set monitor port
  <cr>
(config)#monitor-port monitored 0/12
(config)#monitor-port port 0/11
(config)#exit
#ping 172.16.50.1
Sending 5, 100-byte ICMP Echos to 172.16.50.1, time out is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max 40/40/40/ ms
```

The analyzer did not see the five ping packets that the switch sent when the `ping 172.16.50.1` command was entered. Using the `show port monitor` command (note that the show command reverses the words *port* and *monitor* and leaves out the hyphen), the user discovered that port monitoring was not enabled. After enabling the feature, the analyzer did see the ping commands.

```
#show port monitor
Port monitoring state: Disabled
Monitor port: Ethernet 0/11
Ports being monitored: Ethernet 0/12
#config t
Enter configuration commands, one per line.  End with CNTL/Z
(config)#monitor-port
(config)#exit
#ping 172.16.50.1
Sending 5, 100-byte ICMP Echos to 172.16.50.1, time out is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max 40/42/50/ ms
#
```

CAUTION **Be careful with port monitoring. Port monitoring sends traffic in two directions—to the destined port and to the monitor port. If the monitor port connects more than just a protocol analyzer, this extra forwarding of traffic can cause problems and may result in network loops. We recommend that you use the port monitoring feature only after you have visually verified that nothing other than an analyzer is connected to the monitor (SPAN) port. To avoid loops, some Cisco switches ensure that the monitor port does not participate in the Spanning Tree Algorithm, but don't count on this feature being available on all switches.**

Understanding Network Traffic

As part of your proactive network management, you should use a protocol analyzer to gain an understanding of the day-to-day traffic on your network. Most networks carry user traffic, including terminal/host, client/server, and peer-to-peer application traffic. They also carry background traffic including network management data, exchanges of router topology information, switch and bridge configuration traffic, and polling between servers and devices such as printers.

In today's switched network environments, it is difficult to get a view of network-wide traffic, but as a first step, you should at least study traffic for a typical user and traffic to and from a typical busy file server. First, capture the traffic related to a single workstation as it is booted and as a user logs in and accesses representative applications. Include characteristic user activities in the capture session such as Web browsing, checking e-mail, and printing. This trace file will provide insight into the protocols and configurations that are in use. Second, capture traffic to and from a busy file server. This trace file will provide insight into the mix of protocols that are in use as well as the range of response times experienced by multiple client machines.

Reliable Versus Unreliable Protocols

As you study your network protocols, you will learn that protocols can be reliable or unreliable. A related characteristic is that they can be connection oriented versus connectionless. A reliable protocol is a protocol that has error correction and Positive Acknowledgment with Retransmission (PAR). PAR means that when a device sends data, it expects positive affirmation that the data was received. The device retransmits if it does not receive the affirmation. Senders sequence packets (or bytes, in the case of TCP) so that the recipient can identify which packets (bytes) have been received and which ones are missing. Errors related to lost packets are corrected.

A reliable protocol supports *flow control*, which is a process for adjusting the flow of data from one device to another to ensure that the receiving device can handle all of the incoming data. Flow control is beneficial when a sending device is capable of sending data more quickly than the receiving device can receive the data. With some types of flow control—for example, TCP flow control—the receiver specifies how much data it is able to accept by stating its current *receive window size*.

Unreliable protocols do not have acknowledgments, retransmissions, or flow control. They may have error detection, but no error correction. Unreliable protocols have the advantage of being efficient and easy to implement. They have the obvious disadvantage that they offer only a best-effort service that does not guarantee delivery of data. Unreliable protocols depend on higher-layer protocols to provide reliability.

A connectionless protocol allows a device to transmit data to another device in an unplanned fashion and without prior coordination. Each packet, which is also called a *datagram*, is transmitted independently of previous and subsequent packets. A connection-oriented protocol, on the other hand, has some method for connection establishment and termination. A logical association, or *connection,* is established between devices before any data is transferred. Depending on the protocol, session establishment might be the job of a specific type of device. In some protocols, either side can initiate the session.

When troubleshooting reliable, connection-oriented protocols, verify that sequence numbers, acknowledgments, window sizes, and other parameters associated with this type of traffic are appropriate and are being incremented and managed correctly. If there are multiple retransmissions of segments of data, determine why. Are packets getting lost due to errors, buffer overflows, queuing delays, or other types of congestion? Later chapters will help you answer such questions.

NOTE **Most connection-oriented protocols are reliable, and most connectionless protocols are unreliable, but there are exceptions. For example, Frame Relay is a connection-oriented protocol that does not guarantee packet delivery. Frame Relay requires a virtual circuit to be established before data can be sent, but Frame Relay does not have any reliability mechanisms. In like manner, Open Shortest Path First (OSPF) is connectionless but uses acknowledgments. When a router running OSPF sends link state updates, it sends them as multicast packets in a connectionless fashion, yet it expects acknowledgments from its neighbors.**

In addition to characterizing whether a protocol is unreliable or reliable and connectionless or connection-oriented, it is helpful to classify protocols as supporting one of a few fundamental types of traffic, including terminal/host, peer-to-peer, client/server, server-to-server, and distributed computing network traffic. The next few sections explain these terms to help you gain an understanding of traffic on your network. Subsequent chapters provide more detail on traffic types.

Terminal/Host Traffic

Terminal/host traffic is network traffic caused by a user typing at a terminal or using a terminal emulation program designed to send text-oriented data to a host such as a mainframe or server. The terminal side has little intelligence. The actual application and data are stored on the host. Terminal/host traffic is usually connection oriented. Before data can be sent, a session is established. The amount of data sent by either side of the connection is usually asymmetrical. The terminal sends a few characters and the host sends many characters.

NOTE **Terminal/host traffic could be traffic to a mainframe or traffic to any device, including a router, that supports text-oriented data entered into a simple terminal application. The term *host* sometimes refers to a powerful computing device, such as a mainframe. Networking professionals also use the term *host* in a generic fashion to mean a network device, node, or station. The IP community started this trend of referring to network devices as hosts many years ago.**

Telnet is an example of an application that generates terminal/host traffic. The default behavior for Telnet is that the terminal (user) side sends each character the user types in a single packet. The host returns multiple characters, depending on what the user typed. With some full-screen terminal applications, such as IBM 3270-based terminal applications, the terminal sends characters typed by the user and the host returns data to repaint the screen. The amount of data transferred from the host to the terminal equals the size of the screen plus commands and attribute bytes. Attribute bytes specify the color and highlighting of characters on the screen.

Peer-to-Peer Traffic

Peer-to-peer traffic is generated on networks in which each workstation or process has equivalent capabilities and responsibilities. This differs from client/server architectures, in which some computers are dedicated to serving others. With peer-to-peer traffic, traffic flow is often symmetrical. Communicating entities transmit approximately equal amounts of protocol and application information. There is no hierarchy. Each device is considered as important as each other device, and no device stores substantially more data than any other device.

True peer-to-peer architectures are uncommon. A true peer-to-peer conversation involves the exchange of data in a bidirectional conversation between two machines running peer applications. For example, in a DECnet environment using Network Services Protocol (NSP), two DEC VAX computers might function as protocol-level peers, perhaps in a DEC Local Area VAX Cluster (LAVC) architecture.

In most cases, including the cases that follow, when the term *peer-to-peer* is used, it actually refers to *back-to-back client/server traffic*. With back-to-back client/server traffic, users share resources (files, printers, and so on) from their own machines. The users' machines now play the role of a server. Another user accesses these resources, playing the role of a client. Because any user could be both sharing resources with the network community and using the resources provided by some other member of the network, each machine is both a client and server. Because machines play the same role (both client and server at the same time), they function on an equal basis and are therefore *peers* in the communication environment. When analyzing the traffic with a protocol analyzer, file Input/Output (I/O) protocols would be seen between the machines, but the role of client and the role of server would be played by both machines. To understand the protocols, you should analyze the traffic as two separate client/server conversations.

In small LAN environments, network administrators often set up PCs in a peer-to-peer configuration so that all group members can access each other's data and printers.

There is no central file or print server. On larger networks, peer-to-peer communication is discouraged because it is hard to manage and may result in security problems as users have access to each other's hard drives.

Recently peer-to-peer applications for downloading music, videos, and software have gained popularity. Each user publishes music or other material and allows other users on the Internet to download the data. This is considered peer-to-peer traffic because every user acts as both a distributor and a consumer of data. There is no hierarchy. This type of peer-to-peer traffic should be discouraged on campus networks for two reasons. First, it can cause an inordinate amount of traffic, and, second, the published material is often copyrighted by someone other than the person publishing it. In other words, the person publishing the material is breaking United States copyright law.

Client/Server Traffic

Client/server traffic is generated by a network architecture in which each computer or process on the network is either a client or a server. Servers are powerful computers or processes dedicated to managing disk drives, printers, or other network resources. Clients are PCs or workstations on which users run applications. Clients rely on servers for resources, such as files, peripherals, application software, and processing power. Clients send queries and requests to the server. The server responds with data or permission for the client to send data.

With a client/server architecture, application software runs on the user's machine, which reads and writes data to the disk subsystem in the server. When analyzing client/server traffic, you should observe file I/O commands, replies, and data moving across the network. Client/server protocols include Server Message Block (SMB), Network File System (NFS), Apple Filing Protocol (AFP), NetWare Core Protocol (NCP), and other file I/O protocols.

Client/server protocols are generally reliable and connection oriented. Most modern client/server protocols also include some form of flow control whereby each side of the connection can specify how much data it can receive before the other side should stop and wait for an acknowledgment. Client/server protocols also have methods for a station to temporarily halt data transfer when a recipient experiences congestion.

In a TCP/IP environment, many applications are implemented in a client/server fashion, although the applications were invented before the client/server model was invented. For example, FTP has a client (user) side and a server side. X Windows is an example of a TCP/IP server (the screen manager) that actually runs on the user's machine. This can lead to a lot of traffic in both directions, such as when the user enables a blinking cursor or ticking clock that needs continual updating across the network, even when the user isn't present.

These days, Hypertext Transfer Protocol (HTTP) is probably the most widely used client/server protocol. Clients use a Web browser application, such as Internet Explorer, to talk to Web servers. Each session often lasts just a few seconds because users tend to jump from one Web site to another. Traffic volume is asymmetrical. Clients send small queries and the server responds with large amounts of text and multimedia data. To maximize perceived performance, some Web servers do not send full-sized packets. Packet sizes in the 400- to 600-byte range are typical.

Wireless Clients

When Ethernet connectivity is replaced with wireless connectivity, a *wireless client* is created. A wireless client doesn't know that it's wireless. Applications and upper-layer protocols operate in exactly the same manner as if the client were connected to an Ethernet network. IP still locates a default gateway by sending an Address Resolution Protocol (ARP) broadcast for the gateway's configured address. TCP still enacts a three-way handshake for session setup, sequences and acknowledges data, and provides flow control through the window-size parameter. In fact, the wireless client continues to construct Ethernet frames for carrying Layer 3 protocols. These frames are encapsulated in an 802.11 frame and transmitted through the air via radio frequency (RF) signals. The good news is that using a wireless analyzer, such as WildPackets' AiroPeek, requires the same skills needed for the wired LAN. Analysts must expand their methods and knowledge, however, to include 802.11 protocol behavior and RF signal propagation.

Wireless clients may talk directly to each other (peer-to-peer connectivity) as users share files between notebook computers, for example. Alternately, a wireless client may communicate with a server that is connected to the wired Ethernet using an *access point*. An access point is essentially a Layer 2 bridge between the wired Ethernet and the wireless network. Analyzing wireless network traffic is discussed in full detail in Chapter 4.

Thin Clients

A thin client is designed to be especially simple and to work in an environment where the bulk of data processing occurs on a server. Although the term *thin client* usually refers to software, it is also used for small hardware devices that don't have hard drives. With thin client technology (also known as *server-based computing*), user applications originate on a central server. In some cases, the application runs on the central server; in other cases, the software is installed on the server and is downloaded into the client machine for execution. Microsoft provides thin client solutions with its Windows NT Terminal Server Edition and Windows 2000 with Terminal Services products. Citrix Metaframe is another popular thin client server system. The Citrix solution uses memory in the server to create memory partitions for each client. Software applications run entirely in the Citrix server in the user's partition. The user accesses the software in a terminal/host mode.

An *information appliance* or *computing appliance* is a thin client designed to perform a particular set of dedicated tasks. The promise of computing appliances lies in the concept that dedicated functionality means a system can be easy to use and manage. A computing appliance could be a cash register, a dedicated e-mail machine, or a database retrieval device. Computing appliances often run the Linux operating system and a Java-enhanced Internet browser.

The main advantage of thin client technology is lower support costs. Information Technology (IT) managers can have a centralized base of applications that are managed, configured, and upgraded once, with no need to individually configure each user's machine. In addition, because applications are controlled from the central server, security and intrusion detection can be simplified. Thin client technology is not applicable to every computing application, however, because users may need computers capable of operating without constant connection to a central server.

Server-to-Server Traffic

Server-to-server network traffic includes transmissions between servers and transmissions from servers to NMSs. Servers talk to other servers to implement directory services, to cache heavily used data, to mirror data for load balancing and redundancy, to back up data, and to announce service availability. Servers talk to management applications for some of the same reasons, but also to enforce security policies and to update network management data. Server-to-server traffic may be either connectionless or connection oriented. Flow control may be supported. Traffic volume may be symmetrical or asymmetrical.

Distributed Computing Traffic

Distributed computing attempts to solve a difficult problem by giving small parts of the problem to many computers and then combining the solutions for the parts into a solution for the overall problem. Some complex tasks cannot be accomplished in a reasonable time unless multiple computers process data and run algorithms simultaneously. To make animated movies, for example, designers use multiple computers to speed up graphics rendering. Distributed computing is also used in the semiconductor industry for microchip design and verification, and in the defense industry for military simulations. An unfortunate use of distributed computing is *distributed denial of service* attacks where a hacker marshals many computers to attack a target. The attacker first gains control of many computers by surreptitiously installing a virus or *Trojan horse* via an e-mail attachment or downloaded software. Then, when the time comes to attack the target, the attacker can direct the infected computers to send large amounts of traffic to the target.

Recently, distributed computing projects that make use of computers on the Internet have become popular. Volunteers agree to download software that works in the background, usually when the user isn't doing anything anyway, to help solve a difficult scientific problem. Volunteers all over the world are helping to identify extraterrestrial radio signals, for example. The volunteers' computers accept periodic downloads of data, analyze the data, and send results back to the server on a regular basis. Other Internet distributed computing projects are looking for million-digit prime numbers and helping analyze data in an attempt to fight the AIDS virus. Usually, the Internet computers involved in these projects are home computers. Most campus network engineers discourage participation in these projects at work because of the extra network traffic and the risk that the downloaded software has a virus, Trojan horse, or software bug that may disable the computer or cause network problems.

Statistical Monitoring

In addition to learning about typical protocol behavior, proactive network management involves collecting statistics about network performance. The combined effect of user activity and background traffic results in a particular set of performance statistics on a network. When user activity or background traffic changes, the statistics also change. By monitoring performance statistics, you can establish a baseline of normal behavior. Changes in the measurements may indicate that there are problems

occurring. Problems often show signs of their presence weeks or even months before they impact end users. Statistical monitoring will help you be that experienced guru that we mentioned before who appears to have psychic forewarning about problems.

The following list describes the types of statistical data that you should gather. Many of these terms (such as *bandwidth* and *utilization*) are often misused when discussing network statistics. A goal of this book is to teach you to use these terms correctly:

Bandwidth. The data-carrying capacity of a circuit or network, usually measured in bits per second (bps).

Utilization. The percent of available bandwidth in use.

Throughput. The quantity of error-free data successfully transferred between stations per unit of time (usually seconds).

Accuracy. The amount of useful traffic that is correctly transmitted, relative to total traffic.

Error rate. The number of bad frames or bits compared to good frames or bits, or the number of bad frames in a time period.

Size distribution. A count of frames of different sizes.

Efficiency. A measurement of how much overhead is required to produce a certain amount of data throughput.

Delay (latency). The time between a frame being ready for transmission from a node and delivery of the frame elsewhere in the network.

Delay variation. The extent to which measured delay deviates from average delay.

Response time. The amount of time between a request for some network service and a response to the request.

Active stations. The average number of stations transmitting in a given time period.

Most active stations. A list of the stations that transmit and receive the most data.

When you make statistical measurements, use the data to establish a baseline, to tune your understanding of the network, and to focus troubleshooting efforts. For example, if you assumed that File Server 1 was the most active station, but it turns out that File Server 2 is the most active, then you can avoid wasting time and money trying to optimize and troubleshoot the wrong machine. If it turns out that User 22 is actually the most active station, then perhaps you have a problem. Users may be playing graphics-intensive network games, downloading music or videos, or testing a new nonoptimized piece of software that probably should be tested in the lab first rather than on your operational network.

Analyze the statistics you gather with respect to the network maps and device configurations you also gathered. Understanding the structure of your network and the architecture of devices within the network can help you make sense of the statistics. For example, if you place a 10-Mbps Ethernet protocol analyzer on one port of a switch and configure the switch to mirror traffic for multiple 10-Mbps ports, don't be surprised if the analyzer reports that utilization is 100 percent. The analyzer is simply

computing bits per second received and comparing the rate to 10 Mbps. It has no knowledge of the switch mirroring configuration or switch architecture. A statistic of 100 percent utilization does not mean that your network or your switch backplane are about to go into meltdown, as some novices have been known to report. It's helpful to remember the famous quote attributed to the Victorian-era statesman Benjamin Disraeli: "There are three kinds of lies: lies, damned lies, and statistics."

You should calculate more than a simple mean (average) for some types of statistics. Network performance data is often bimodal or heavily skewed from the mean. For example, response time from a server is often bimodal, if the server sometimes retrieves data from speedy Random Access Memory (RAM) cache and sometimes gets the data from a slow mechanical disk drive. When network performance data is bimodal, multimodal, or skewed from the mean, you should document a standard deviation with any measurements of the mean. Standard deviation is a measurement of how widely data disperses from the mean.

Simple Network Management Protocol and Remote Monitoring

One method for capturing statistical and troubleshooting data is to use a standard network management protocol such as SNMP. SNMP is a language for retrieving data from a managed device and setting configuration options on the device. An *SNMP agent* is software that resides in a managed device that keeps a database of management information in the device. SNMP uses the term *Management Information Base* (MIB) for a database of management data. An SNMP NMS polls the agent and runs applications to display management data.

The RMON MIB was developed by the IETF in the early 1990s to address shortcomings in the standard SNMP MIBs, which lacked statistics on data link and physical-layer parameters. The IETF originally developed the RMON MIB to provide Ethernet traffic statistics and fault diagnosis. In 1994, Token Ring statistics were added. Chapter 3 discusses Ethernet RMON in more detail.

SNMPv2, which is defined in Request for Comments (RFC) 1905, describes seven types of packets used for retrieving management data and setting parameters on SNMP and RMON agents:

- *Get Request.* Sent by an NMS to an agent to collect a management parameter.
- *Get-Next Request.* Sent by an NMS to collect the next parameter in a list or table of parameters.
- *Get-Bulk Request.* Sent by an NMS to retrieve large blocks of data, such as multiple rows in a table (not in SNMPv1).
- *Response.* Sent by an agent to an NMS in response to a request.
- *Set Request.* Sent by an NMS to an agent to configure a parameter on a managed device.
- *Trap.* Sent autonomously (not in response to a request) by an agent to an NMS to notify the NMS of an event.
- *Inform.* Sent by an NMS to notify another NMS of information in a MIB. Supports Managers of Managers (MoM) architectures (not in SNMPv1).

WildPackets Statistical Tools

On many campus networks, devices are not configured to use SNMP or RMON. In these cases, and in situations where SNMP and RMON do not provide the desired data in an easy-to-use fashion, a portable protocol analyzer is a better option. WildPackets' EtherPeek and AiroPeek products are best known for their packet capture and protocol analysis features, but they are also valuable tools for statistical monitoring. EtherPeek captures global statistics based on all network traffic from the moment the NIC is chosen. It also captures statistics based only on the unhidden packets in the capture buffer or on a user-controlled sample of network traffic.

EtherPeek and AiroPeek display real-time packet counts and traffic volume for the network as a whole and for each node on the network. In addition, the Protocol Statistics window shows network traffic volume broken down by protocol and subprotocol, which is essential information when learning about and optimizing a network. The Conversation Statistics window shows traffic statistics aggregated by source/destination node pairs, and by protocol. The Network Statistics window displays a real-time utilization percentage and a packets-per-second rate. The Error Statistics window shows error counts, and the Size Statistics window shows the distribution of packet sizes.

WildPackets also provides useful tools for application-layer statistical monitoring. The WebStats Analysis Module, for example, adds Web site management tools to the EtherPeek program. It collects data on TCP-based protocols, including HTTP and FTP data streams, and displays the results in EtherPeek.

Simulation and Modeling

Simulation and modeling software is another ingredient in a toolbox of proactive network management applications. This type of software typically uses object-oriented data structures to help you predict the performance of a network after a planned reconfiguration or redesign or to help you stress-test a model of your current network. Simulation and modeling tools let you select objects that represent network topology, protocols, traffic levels, and routing algorithms to simulate the operation of a network. The output provides network performance predictions, including response times, throughput measurements, network utilization, and packets dropped by routers.

For a simulation tool to be effective, it must be developed by software engineers who understand computer networking in addition to statistical analysis and modeling techniques. Because performance problems on networks often arise from the complex interaction of media access methods, switch and router architectures, server architectures, and software implementations of buffers and queuing on these devices, modeling network behavior can be challenging. One solution to this problem is that a simulation tool can incorporate measurements of actual network traffic, rather than relying solely on device libraries that model theoretical behavior. This approach not only solves the problem of modeling complex devices, but also allows the tool to calibrate assumptions made about traffic load and characteristics. There is less reliance on the user of the tool to accurately predict traffic load, and more reliance on real measurements. A recommended tool that uses this calibrated approach is NetPredictor from a company called NetPredict.

Active and Reactive Troubleshooting

No matter how much proactive monitoring and optimization you do, your network is going to experience failures at times, requiring you to react to problems. Using a systematic method, such as the Cisco troubleshooting method or some other orderly process, you will need to define, isolate, and fix the problem. There are a variety of tools to use when reacting to problems. In addition to some of the tools already mentioned, which can be used for both proactive and reactive troubleshooting, you can also use the tools described in the next few sections.

Tools for Troubleshooting the Physical Layer

Many tools are available for troubleshooting at the physical layer. Volt-ohm meters and digital multimeters are low-end devices that can provide basic troubleshooting information. These devices measure electrical indicators such as alternating current (AC) and direct current (DC) voltage, current, resistance, capacitance, and cable continuity. Although you may be asked a question about these basic tools on the Cisco Support Test, in actual field practice most network support engineers test with more sophisticated tools.

Cable testers (also sometimes called *scanners*) are useful tools for testing cables and checking physical connectivity. Cable testers are available for Shielded Twisted Pair (STP), Unshielded Twisted Pair (UTP), and coaxial and twinax cables. These testers can test and report cable conditions including Near-End Crosstalk (NEXT), attenuation, and noise. Some of the tools also have a Time-Domain Reflectometer (TDR) function, wire-mapping features, and traffic-monitoring capabilities. Some testers display MAC-level information about network utilization, error rates, and collisions. Some tools also allow for limited protocol testing—for example, sending IP pings.

Similar tools exist for fiber optic cabling. Because of the relatively high cost of fiber optic cable and its installation, it is recommended that fiber optic cable be tested before installation (also called *on-the-reel testing*) and after installation. Continuity testing of the fiber requires either a visible light source or a reflectometer. Light sources capable of providing light at the three predominant wavelengths—850, 1300, and 1550 nm—are used with power meters that test attenuation and return loss in the fiber.

A TDR locates kinks, sharp bends, shorts, opens, impedance mismatches, and other defects in copper cables. Optical TDRs (OTDRs) work on fiber optic cabling. A TDR works by bouncing a signal off the end of the cable, much like radar. Opens, shorts, and other problems reflect the signal back at different amplitudes, depending on the problem. A TDR measures how much time it takes for the signal to reflect. Together with knowledge of the speed at which a signal travels in a cable, a measurement of how much time it takes for the signal to reflect gives the TDR an estimate of the distance to the problem. TDRs can also be used to measure the length of a cable. When a signal reaches the end of a cable, it reflects at a very low amplitude.

When troubleshooting physical-layer interfaces, it sometimes helps to measure digital signals that are present. A breakout box is often attached at a data communications interface—for example, between a computer and a modem or between a router and a Channel Service Unit (CSU). A breakout box monitors interface signals and displays information using Light-Emitting Diodes (LEDs). For example, an EIA/TIA-232 breakout box might

contain 25 LEDs, 1 for each signal on the DB-25 EIA/TIA-232 connector. A positive voltage might cause the LED for a signal to light; a negative voltage might cause it to turn off. The user can monitor the LEDs to troubleshoot control signals.

Some testing tools have the capability to perform a Bit Error Rate Test (BERT). With BERT, a known pattern of 1s and 0s is continually transmitted across a link so that proper reception can be measured at the other end. If 1000 bits are transmitted and two bit errors are detected, for example, the bit error rate for that circuit is 0.002. Another measure of reliability, the Block Error Rate Test (BLERT), measures the ability of a link to pass entire blocks of data rather than just bits. When using most modern protocols, a BLERT test may be more indicative of actual reliability, because protocols retransmit a whole block of data if a single bit error occurs.

Tools for Troubleshooting Above the Physical Layer

This section focuses on commands and tools you can use to quickly test reachability to a remote device or network, determine the path to a remote device, determine name-to-address mapping, and efficiently test other upper-layer functions. Theoretically, these tools should be used after the physical-layer tools mentioned in the previous section have verified that the physical layer is functioning correctly. In actual field practice, however, these tools are usually used before testing at the physical layer. If the tools indicate a problem, then the troubleshooter reverts to a check of the physical cabling and interfaces before deciding that the problem lies at a higher layer. Troubleshooting is an iterative process that migrates between OSI layers as the troubleshooter gathers facts about the problem.

Ping

The *Packet Internet Groper (ping)* utility is a popular and effective tool for testing reachability to a remote device. Ping sends a command to a remote station that causes the station to respond, somewhat like a Ping-Pong ball returns when it is sent across a net to a partner. Ping is usually used to test reachability to an individual device. *Ping scan* sends to a range of addresses. The intended purpose of ping scan is to discover active nodes on a network during a network reconfiguration when new machines are being added or removed from the network. Ping scan is sometimes misused by hackers.

Although ping originated in the TCP/IP community, Cisco allows the use of the Cisco IOS ping command to test reachability to devices running the following protocols:

- Apollo
- Connectionless Network Service (CLNS)
- DECnet
- Banyan Virtual Integrated Network Service (VINES)
- Xerox Network System (XNS)
- IP
- AppleTalk
- Novell Internetwork Packet Exchange (IPX)
- SNA

The SNA ping requires you to type the whole command `sna ping`. (For the other pings, you type `ping` followed by `enter` and then select the protocol.) The SNA ping initiates an Advanced Program to Program (APPC) session. The AppleTalk ping is an AppleTalk Echo Protocol (AEP) packet and is covered in more detail in Chapter 11. The IPX ping frame format depends on configuration parameters and is covered in more detail in Chapter 10. The IP ping is actually an Internet Control Message Protocol (ICMP) echo packet. A detailed view of a packet capture of an IP ping (echo) reply is shown in Figure 2.4.

When in privileged mode on a Cisco IOS router, there are many options for the ping command. Enter `ping` with no parameters and immediately press `enter` to see the options. One useful option is the selection of the source address for use in the ping packet. Because a router has potentially many IP addresses, setting the source address for the packet is a useful troubleshooting feature. If you don't set the source IP address, the router uses by default the address of the outgoing interface used to reach the device you are trying to ping. When routing problems are occurring, however, the router can be confused regarding which outgoing interface to use. In this case, being able to explicitly configure the source address is helpful. The following example shows the use of ping options.

```
Charlotte# ping
Protocol [ip]:
Target IP address: 172.16.40.2
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: yes
Source address: 172.16.10.1
Type of service [0]:
Set DF bit in IP header? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.40.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent, round-trip min/avg/max = 40/40/40 ms
```

When pinging on a Cisco router or switch, the default is to send five ping packets. The router displays five characters, one for the result of each ping. The five exclamation points in the example indicate that five pings were successfully sent and a reply was received for each one.

A ping, which is really an ICMP echo packet, results in an ICMP echo reply when there are no problems. If there is a problem, the result is often an ICMP error message. The error message can come from a router en route to the destination or from the destination itself. ICMP packets have a type field and a code field. The combination of the type and code provides the significance of the message. Table 2.2 lists ICMP types and codes relevant to ping and to trace-route, which is covered in the next section.

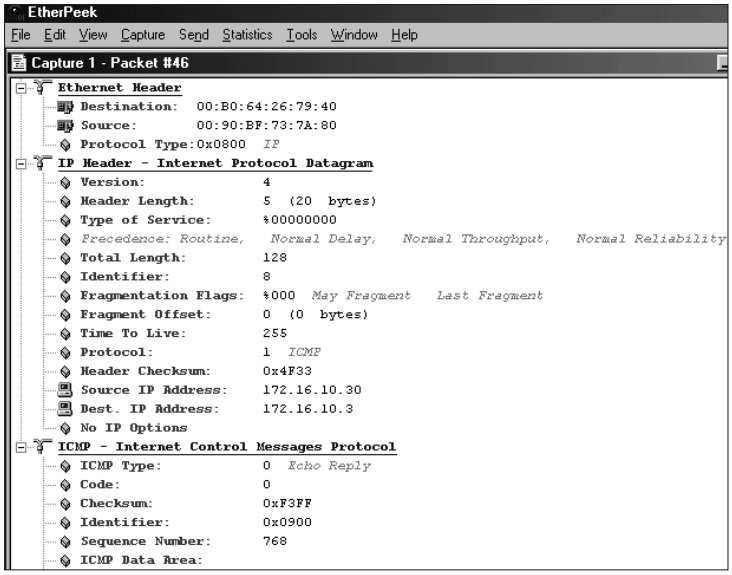


Figure 2.4 A Ping reply packet.

Table 2.2 ICMP Types and Codes

TYPE	CODE	MEANING
0	0	Echo reply (ping reply).
3	x	Destination unreachable (generic category).
3	0	Network unreachable.
3	1	Host unreachable.
3	2	Protocol unreachable.
3	3	Port unreachable.
3	4	Fragmentation was needed and the Don't Fragment (DF) bit was set.
3	5	Source route failed.
3	13	Packet administratively prohibited.
4	0	Source quench.*
5	x	Redirect (generic category).
5	0	Redirect datagrams for the network.
5	1	Redirect datagrams for the host.

(continues)

Table 2.2 ICMP Types and Codes *(Continued)*

TYPE	CODE	MEANING
5	2	Redirect datagrams for the type of service and network.
5	3	Redirect datagrams for the type of service and the host.
8	0	Echo (ping).
11	x	Time exceeded (generic category).
11	0	Time-to-Live (TTL) exceeded.
11	1	Fragment reassembly time exceeded.
12	0	Parameter problem.

*Per RFC 1812, a router should not originate source quench messages. Per RFC 1122, however, a host may send source quench messages.

Cisco routers and switches output a character code that represents the received message and the result of the ping. Table 2.3 shows the character codes for ping. Cisco IOS software does not provide detailed information on the ICMP error message received when using ping. In most cases, if there is a problem, Cisco IOS software simply outputs a period, meaning there was no response, or a U, meaning an ICMP destination unreachable message was received. Cisco IOS software does not distinguish between network unreachable, host unreachable, or protocol unreachable when using ping.

Table 2.3 Cisco IOS Ping Result Codes

RESULT CODE	MEANING
!	Each exclamation point indicates receipt of an ICMP echo reply.
.	Each period indicates the router timed out while waiting for a reply.
U	A destination unreachable message was received.
Q	A source quench message was received.
M	A fragmentation needed and DF bit set message was received.
?	An unknown packet was received.
C	A packet was received with the congestion-experienced bit set.*

* See RFC 2481, an experimental protocol that adds explicit congestion notification to IP.

To prevent an errant packet stream from impacting router performance, Cisco routers limit the rate at which they send ICMP destination unreachable messages. So when a problem with pinging occurs, you probably won't see UUUUU in a network with Cisco routers. You will most likely see U.U.U. Every other ping times out without an explicit error. In Cisco IOS version 12.1 and later, the rate limiting is configurable with the `ip icmp rate-limit unreachable` command.

Trace-Route

Cisco IOS software and most operating systems have a *trace-route* facility for investigating the routing path to a destination device. Trace-route displays the sequence of hops a packet traverses to get from a given source to a destination. Trace-route originated in the TCP/IP community, but Cisco lets you use the tool for other protocols also, including AppleTalk, CLNS, Novell IPX, and Banyan VINES.

The results provided by trace-route are a measurement of the round-trip time to each router in the path to a destination and also a measurement of the round-trip time to the actual destination. The timing measurements account for processing time at the recipients in addition to propagation delay. Trace-route can be used as a rough estimate of delays on a network. It is most useful, however, as a method for determining the path to a remote destination.

With UNIX and Cisco IOS operating systems, an IP trace-route packet is a User Datagram Protocol (UDP) *probe* sent to a high UDP port number, usually in the 33,000 to 43,000 range. Trace-route works by taking advantage of the ICMP error message a router generates when a packet exceeds its TTL value. TTL is a field in the IP header of an IP packet.

Trace-route starts by sending a UDP probe packet with a TTL of 1. This causes the first router in the path to discard the probe and send back a TTL exceeded message. One of the first things a router does when forwarding IP packets is decrement the TTL (which is essentially a hop count value). If the decrement causes the TTL to reach 0, then the packet is dead (discarded) and a TTL exceeded message is sent.

The trace-route command sends several probes, increasing the TTL by 1 after sending three packets at each TTL value. For example, trace-route sends three packets with TTL equal to 1, then three packets with TTL equal to 2, then three packets with TTL equal to 3, and so on, until the destination host is reached or a configured maximum number of tries (usually 30) is reached.

Each router in the path decrements the TTL. The router that decrements the TTL to 0 sends back the TTL exceeded message. The final destination host sends back a port unreachable ICMP message, because the high UDP port number is not a well-known port number. This process allows a user to see a message from every router in the path to the destination, and a message from the destination.

NOTE When executing a trace-route to a Cisco router, keep in mind that the sending of port unreachable messages may be rate limited.

The trace-route facility in Microsoft operating systems sends a ping rather than a UDP packet. The trace-route command makes use of the IP TTL feature and router behavior